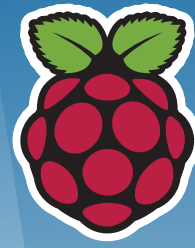


ПРОВЕРЕНО: 5 ЛУЧШИХ ЗАПУСКОВ

LINUX FORMAT

Главное в мире Linux

Апрель 2014 № 4 (182)



Выпечка Pi

» Узнайте, как Sony штампует по 10 000 Raspberry Pi в день **с. 50**

НА ВАШЕМ БЕСПЛАТНОМ DVD
Tiny Core, Linux в 12 МБ и сверх того!
Tails Live, Gentoo,

ЗАБЬЕМ АНБ

Защитите свои данные и личную жизнь — и на настольном ПК, и в облаках

- » По Сети — анонимно
- » Безопасная личная почта
- » Шифрование как у военных



ПЛЮС!
Языки программирования доведут до...
с. 76+



Король Ubuntu

«Следуя за толпой, навсегда останешься посредственностью»

Марк Шаттлворт о спорах и бизнесе

Также в номере...

Безопасный Android
» Превратите свой мобильник в цифровую твердыню **с. 24**



Raspberry Pi
Торрент-сервер
» Строим выделенный сервер

Коллекция книг
Крупного Calibre
» Делимся библиотеками всей семьей

Wine
Silverlight в Linux
» Недруг изолирован! Поток вещает Pipelight

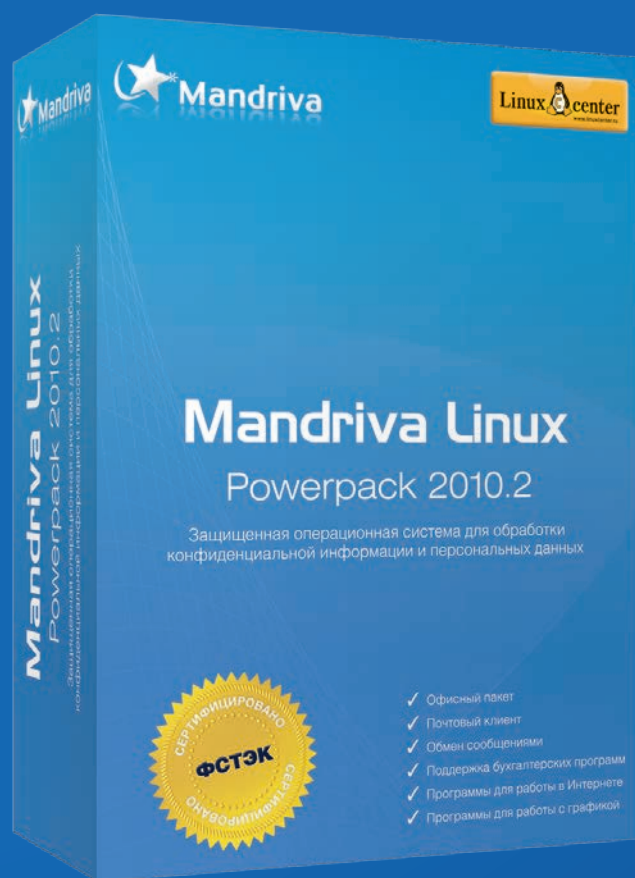
ПОДПИСНЫЕ ИНДЕКСЫ В КАТАЛОГАХ
Агентство «Роспечать» — 36343,
«Почта России» — 11932, «Пресса России» — 90959

Linux center
www.linuxcenter.ru

Mandriva Linux

Powerpack 2010.2

Дистрибутив может использоваться для обработки конфиденциальной информации в автоматизированных системах класса до 1Г включительно и обработки персональных данных в информационных системах класса до К1 включительно.



Простота
установки, работы
и администрирования

Легкость
освоения новичками
в GNU/Linux

Широчайший спектр
поддерживаемого
оборудования

Цена 4990 руб.

Дистрибутив Mandriva Linux Powerpack 2010.2
прошел сертификацию ФСТЭК и предназначен для создания
защищенной офисной рабочей станции или небольшого сервера.



www.linuxcenter.ru/shop/certified_fstek/mandriva_fstek/mandriva_2010-2_pp_certific_fstek/

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Что мы делаем

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



Кто мы

В этом номере мы рассматриваем, как сохранить анонимность в Сети. И мы спросили наших авторов: какие методы они используют для самозащиты?



Крис Торнетт

Залускаю сеанс Tails в виртуальном Privoxy на компьютере, который потом сжигаю.



Мэтт Хэнсон

Использую для всех коммуникаций почтовых голубей, а потом стираю их память.



Эфраин Эрнандес-Мендоса

У меня такой сложный код шифрования, что я и сам его не понимаю.



Лес Паундер

Подобно Ричарду Столлмену, я избегаю web-браузеров и всю дорогу использую протокол SSH.



Маянк Шарма

Непренемно использую Tor — переделал Raspberry Pi в постоянно действующий Tor-прокси.



Нейл Ботвик

Я нанял могучего мастера-чародея, чтобы он закодировал мой раздел /home. Да у него еще и пароль имеется.



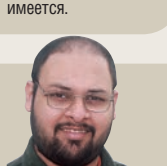
Джо Джордан

Хижину в горах с сигнализацией на базе Raspberry Pi против грабителей/дронов. Плюс жестяной колпак.



Джульетта Кемп

Я пользователь GnuPG, и общаюсь только с теми людьми, которые могут читать мою личную почту.



Шашанк Шарма

Когда я в чате онлайн, я использую OTR и перед каждой репликой говорю «Не записывать!»



Дэвид Хейвард

Избегаю интернет-кафе и точек публичного доступа Wi-Fi как чумы и использую Tor где только можно.



Марко Фиоретти

Если я расскажу вам про все свои меры предосторожности, мне придется вас убить. Извините.



Валентин Синицын

Я регистрируюсь в соцсетях под именем своего начальника, а в качестве аватары ставлю котеночка.



«Дивный новый мир»

Историки техники утверждают, что в развитии любой технологии найдется изобретение, дающее старт массовому использованию. К примеру, для начала всеобщей автомобилизации был необходим электрический стартер, позволивший сесть за руль даже хрупким дамам. В развитии Интернет таким изобретением стала «Всемирная Паутина» — URL, HTML и HTTP — идея которой была впервые высказана 25 лет назад, в марте 1989 года.

Именно WWW превратила Сеть из «игрушки для гиков» в систему, полезную «для домохозяек». Как современник этой революции (пользуюсь Инетом с весны 1991-го), могу утверждать, что различие в функциональности и удобстве «до» и «после» примерно такое же, как между телетайпом и современным сотовым телефоном.

В дальнейшем, с появлением блогов, социальных сетей и поисковых машин, человечество получило доступ в новое измерение, которое иногда называют киберпространством. Мир без расстояний и границ для обмена информацией.

Но, как это случилось с открытыми ранее новыми измерениями, например воздушно-космическим, быстро нашлись желающие в этом измерении воевать. По всем канонам. Со средствами нападения, защиты, разведки и, естественно, маскировки, которые и стали темой этого номера.

Кирилл Степанов

Главный редактор

» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

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

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

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

9-10 АПРЕЛЯ | САНКТ-ПЕТЕРБУРГ
В РАМКАХ «ПРЕМИИ HR-БРЕНД 2013»

X ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ ПО HR-БРЕНДИНГУ

СПЕШИТЕ ЗАБРОНИРОВАТЬ МЕСТА:

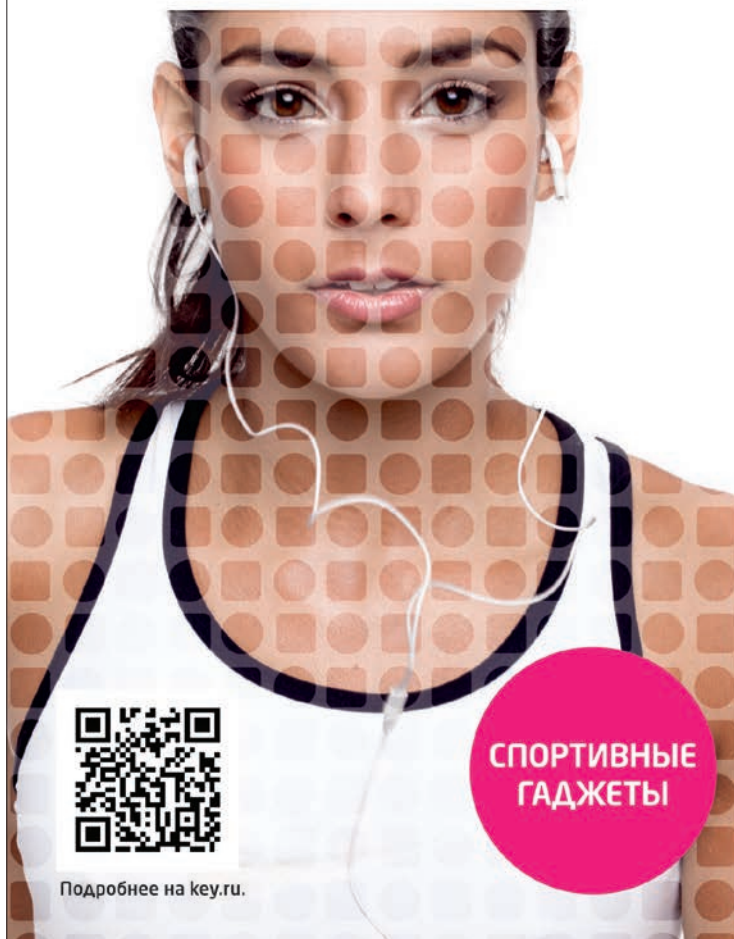
звоните своему персональному менеджеру НН

(+7 812 607 77 29)

или пишите по адресу hrconf@hh.ru



ТВОИ ПОМОЩНИКИ В ТРЕНИРОВКАХ



Подробнее на key.ru.

СПОРТИВНЫЕ
ГАДЖЕТЫ

Экшн - камера



Экшн-камера Sony HDR-AS30

Smart - часы



Smart-часы Cookoo Watch

Smart - здоровье



Браслет Jawbone UP



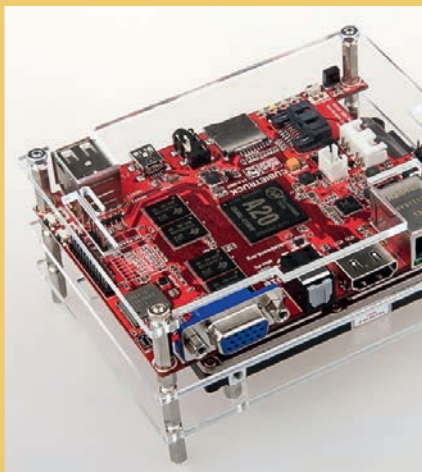
Шаромер Misfit Shine

Товар сертифицирован. Количество товара ограничено.



Одноплатный компьютер Cubieboard3

Cubietruck



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- Процессор: AllWinnerTech SOC A20, ARM Cortex-A7 Dual-Core
- Видео: ARM® Mali 400 MP2
- Выходы: HDMI, VGA 1080p
- Оперативная память: 2 Гб, DDR3, 480 МГц
- Встроенная память: 8 Гб NAND
- Внешние накопители: интерфейс SATA 2.0 для 2,5" HDD; возможно подключение 3,5" HDD с использованием внешнего питания 12 В
- Сеть: 10M/100M/1G Ethernet, Wi-Fi и Bluetooth
- Питание: DC 5 В, 2,5 А с подключенным HDD; возможно подключение

литиевого элемента питания

- Периферия: 2 × USB host, 1 × USB OTG, 1 × Toslink (SPDIF), 1 × IR, 4 × LEDs, 3,5" аудиовыход, 3 кнопки
- Поддерживаемые ОС: Android, Linux-based, BSD
- 54 контакта расширения I2S, I2C, SPI, CVBS, LRADC × 2, UART, PS2, PWM × 2, TS/CSI, IRDA, LINEIN & FMIN & MICIN, TVIN × 4
- Размер платы: 11 × 8 см

КОМПЛЕКТАЦИЯ

- Плата Cubietruck
- Прозрачный акриловый корпус
- Кабель питания USB
- Кабель MiniUSB
- Кабель SATA с питанием
- Радиатор для процессора

3800
руб.

www.linuxcenter.ru/shop/ops_hard/cubieboard/cubietruck/

«ГНУ/Линуксцентр»: +7 812 309-06-86 (Санкт-Петербург), +7 499 271-49-54 (Москва)



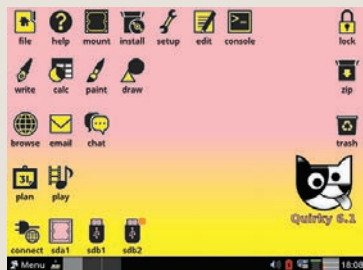
Содержание

Напечатано в совершенно секретном бункере не скажем где.

Обзоры

Quirky 6.1.1 14

Щенки и ветки обычно не слишком ладят, но данная ветвь Purru Linux запросто воскресит ваше старое железо.



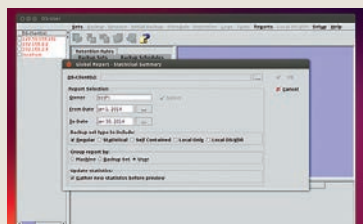
» Quirky прищиплет трудоустройство вашим машинам-пенсионерам.

Securstore Cloud 15

Сервис облачного хранения класса предприятия. Насколько он быстр и надежен?

Fritzing 0.8.5b 16

Проектируйте собственные монтажные схемы — и наконец-то обзаведетесь вожделенной армией роботов. Ведь вы этого заслуживаете!



» С Fritzing 0.8.5b изготавливать самопальную электронику проще простого.

phpMyAdmin 4.1.5 17

Мы протестировали свежую версию инструмента администратора MySQL.

NAS4Free 18

Добиваемся полного контроля над своим сетевым устройством хранения.

ioSafe 214 19

Жгите его огнем, заливайте водой — данные в этом NAS устоят под пытками.

To The Moon 20

Луна или утренний снег... / Любуясь прекрасным, я жил, как хотел. / Вот так и кончаю год.

ЗАБЬЕМ АНБ

Защитите свою конфиденциальность, не позволив следить за собой в Сети с. 34



Сравнение: Пускачи приложений с. 28



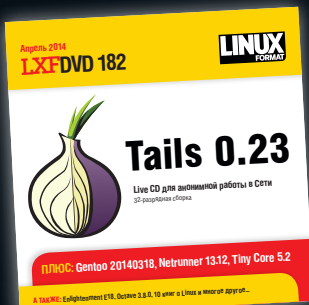
Люди говорят



« Есть мнение, что Linux — нечто экзотическое, подтверждение выдающегося ума »

Марк Шаттлворт *Полемика и Ubuntu* с. 46

На вашем бесплатном DVD



Linux Tails

» Live-диск супер-защищенного дистрибутива

Netrunner

» Мощный потомок Ubuntu

ПЛЮС: HotPicks и коды учебников **с. 106**

Ищите в этом номере



Как оградить Android 24

Защищены ли данные на вашем мобильнике?

Станем профи Linux 42

Как начать карьеру Linux-профессионала.

Выпечка Raspberry Pi 50

Мы посетили фабрику, где выпускают Pi.



Академия кодига

PHP 76
Как использовать этот язык, чтобы наскрести данных с сайтов.

OCaml 80
1970-е ознаменовались началом движения панков и языком программирования ML. Язык OCaml — его прямой потомок.

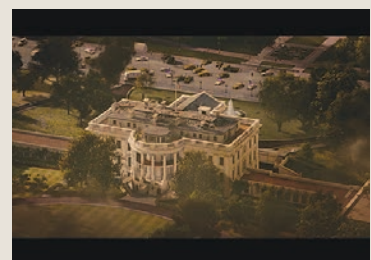
Lua 84
Представляем язык скриптов с супер-облегченным движком, удобный для встраивания в приложения.

Squeak 88
Объектно-ориентированный язык, замечательно приспособленный для образовательных целей.

Erlang 92
Собственно, это не совсем про Erlang, а про общие принципы тестирования кода.

Учебники

Плагин видео Pipelight 62
Плагин для web-браузера, альтернатива Silverlight с открытым кодом.



» Pipelight: смотрите Netflix в Linux.

Gentoo Знакомимся 64
Дистрибутив с высочайшей настраиваемостью — проходим от установки до командной строки.

Управление e-книгами Calibre 68
Беллетристика и справочники — все вперемешку? Презираете десятичную систему Дьюи? Ай, нехорошо. Вам следует «калибровать» свою библиотеку...

Торренты Машина для торрентов 72
Незачем сидеть, уставясь в прогресс-индикатор загрузки, который черепашими темпами ползет к 100%. Берите старый ПК (или вообще Raspberry Pi) — и превратите его в постоянно действующий торрент-сервер.

Постоянные рубрики

Новости 6
Mozilla блокирует плагины, Fedora вводит частичную анонимность, патенты на ПО признаны злом, Tor защищает сообщения, сервис Persona отдал сообществу, Gnome хочет портироваться, множатся телефоны с Firefox OS, а Китай ни с того, ни с сего спускает Red Flag.

Новости Android 22
Видеозвонки — это уже не из другой Oregа. Яндекс выпустил прошивку для смартфонов, Android попал на игровую консоль, а планшет MediaPad X1 выбил в претенденты на звание лучшего.

Сравнение 28
Мы протестировали пять лучших программ для запуска приложений, которые сэкономят ваше драгоценное время и заставят Unity и Gnome Shell дрожать за свои лавры.

Интервью LXF: Мир, как его видит Марк 46
Марк Шаттлворт считает, что в мире, где все мнения равны, добиться чего-нибудь невозможно.

Что за штука... Tor 54
Нас не догонят! Выходим в Интернет через эту анонимную сеть.

Рубрика сисадмина 56
Доктор выкладывает всю подноготную ядра и выражает скептицизм насчет партнерства Red Hat и CentOS.

Ответы 96
ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!
Нейл Ботвик — про неработающий автологин, переименование нескольких файлов, принуждение компилятора к работе, первую попытку использования Linux, проблемы при компиляции из исходников, резервное копирование с rsync.

HotPicks 100
Отдайте горяченького! Лучшие в мире новинки свободного ПО: abc2midi, Brackets, DevCoin, Enlightenment E18, GnuCash, GNU Grep, GNU Octave, Hnefatafl, MKVToolNix, PlaneShift, Shell AudioScope.

Диск Linux Format 106
Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108
Не беда! Но поспешите его купить, пока он не выставлен на аукционе дорогостоящих раритетов.

Через месяц 112
Перестрелка дистрибутивов: не на жизнь, а на смерть! Плюс: 3D-печать и соревнование сред разработчи.

ПЛАГИНАМ — БОЙ

Firefox без плагинов

Mozilla меняет политику в отношении плагинов.

Первым этапом в реализации плана полного блокирования всех плагинов в *Firefox* станет отображение контента только по щелчку. Это решение уже было апробировано в 26-й версии браузера; правда, там оно работало только с устаревшими версиями NPAPI-плагинов. В будущем предполагается распространить метод на все встроенные модули, кроме добавленных в белый список.

Выбор разработчиками пути свидетельствует о том, что быстро отказаться от плагинов не получилось, несмотря на то, что именно они являются причиной главных проблем со стабильностью, безопасностью и производительностью. Теоретически все проблемы должны были решиться переходом от использования плагинов к применению web-технологий. Но на практике разработчики к этому пока

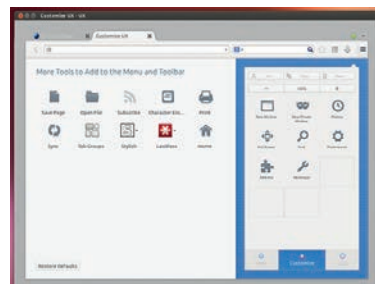
не готовы. Отсюда и компромиссный подход, который должен дать разработчикам дополнительное время. Однако затягивать тоже нехорошо, поэтому условия для помещения плагина в белый список достаточно строги.

Во-первых, автор плагина должен будет предоставить план перехода с NPAPI-плагинов на стандартные web-технологии. Во-вторых, это следует сделать до 31 марта — иначе плагин не успеет пройти должного тестирования (кстати, на всё время нахождения плагина в белом списке его авторы обязаны выполнять тесты, связанные с контролем качества). В-третьих, плагин заносится в белый список только на четыре версии *Firefox*, а потом всю процедуру следует проходить заново.

Если принять во внимание достаточно высокую рыночную долю браузера *Firefox*,

то у авторов плагинов по сути не остаётся иного выхода, кроме реализации требуемой функциональности стандартными методами. В противном случае технология может оказаться недоступной пользователю со всеми вытекающими из этого последствиями.

Ситуация усугубляется тем, что меры по прекращению поддержки плагинов планирует принять компания Google — уже в ближайшей версии браузера *Chrome* для Linux будет реализован оригинальный интерфейс PPAPI. Таким образом, в ближайшем обозримом будущем плагины будут вытеснены стандартными web-технологиями. И разработчикам порталов следует с этим смириться.



» Из браузера *Firefox* скоро будет удалена поддержка плагинов.

ПОРА ПРЯТАТЬСЯ

Don't Ask Don't Tell

Сообщество разработчиков Fedora вводит частичную анонимность.

К сожалению, политика всё чаще вторгается во все сферы нашей жизни. В том числе и в разработку свободного ПО. В частности, согласно законам США компания Red Hat не имеет права экспортировать свою продукцию в страны так называемой «Оси Зла», к которым относятся Куба, Иран, Северная Корея, Судан и Сирия. Это требование распространяется и на Fedora Community.

Разумеется, такое условие приводит к тому, что из сообщества разработчиков автоматически исключаются граждане некоторых стран. Причём независимо

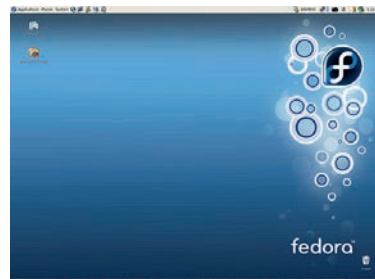
от их личного отношения к политике. Очевидно, это нравится не всем. Но напрямую нарушать закон — тоже не вариант.

Поэтому сообщество Fedora Community решило бороться с дискриминацией иными способами. Они решили больше не указывать своё гражданство во всех внутренних коммуникационных системах. И вопрос о чём-либо гражданстве в сообществе теперь является недопустимым.

Подобный метод называется “Don't Ask Don't Tell [Не спрашивай и не говори]” и применялся он военнослужащими армии США, отличающимися нетрадиционной

сексуальной ориентацией. И показал на практике свою эффективность.

Разумеется, если разработчик всё-таки решил указать своё гражданство, препятствовать ему не будут. Но делает это он на свой страх и риск: если его страна попадёт в «чёрный список» правительства США, то по закону его исключат из общества. А если гражданства не указывать, то никаких оснований для исключения не будет, как и претензий к Fedora Community со стороны властей.



» Сообщество разработчиков Fedora предлагает участникам скрывать своё гражданство.

ПРОТИВ ПАТЕНТОВ

Общая проблема объединяет

В области патентов мнение FSF, SFLC и OSI полностью совпадают.

Общезвестно, что открытое и свободное ПО — не одно и то же. Сторонники первого считают, что доступность исходного кода способствует повышению эффективности разработки. Поклонники второго предпочитают говорить о праве каждого на распространение и модификацию программ. Но относительно патентов на ПО их точки зрения совпадают полностью.

Недавно FSF (Free Software Foundation), SFLC (Software Freedom Law Center) и OSI (Open Source Initiative) совместно выступили в качестве независимого участника в судебном разбирательстве между компаниями Alice и CLS, речь в котором шла как раз о патентах на программное обеспечение. При этом подчёркивалось, что они никак не заинтересованы в победе одного из участников, но заинтересованы в принятии судом правильного решения.

Именно по этой причине организациям и потребовалось участие в процессе, предоставляющее им право на предоставление суду собственного экспертного заключения. То есть — уже совершенно официального документа, с которым суд обязан считаться.

В заключении прямо говорится, что лежащие в основе любого ПО идеи никоим образом не относятся к разряду подлежащих патентованию сущностей, поскольку не являются ни «механизмом»,

«Свободный обмен идеями только способствует инновации.»

ни «преобразованием». Они не привязаны к конкретному физическому устройству и не позволяют преобразовывать предметы на физическом уровне. Таким образом, со строго юридической точки зрения само понятие патента на ПО лишено всякого смысла. Не говоря уже о том, что патенты на ПО напрямую нарушают Первую поправку к Конституции США.

В качестве справочной информации в экспертном заключении приводится тот факт, что на ранних этапах развития отрасли отсутствие патентов на ПО никак не сказывалось на темпах роста. Напротив, свободный обмен идеями только способствует инновациям и препятствует

появлению монополий. Стало быть, отмена патентов отразится на индустрии только положительно.

В пример эксперты привели свободное и открытое ПО. Несмотря на отсутствие патентования, этот сегмент развивается весьма динамично и успешно. Успешно настолько, что свободные решения используют компании, являющиеся крупнейшими держателями патентов, что само по себе выглядит достаточно странно.



Free Software Foundation продолжает активно выступать против патентования ПО.

БЕЗОПАСНЫЙ ИНТЕРНЕТ

Tor Instant Messaging Bundle

Проект Tor создаёт защищённый сервис мгновенного обмена сообщениями.

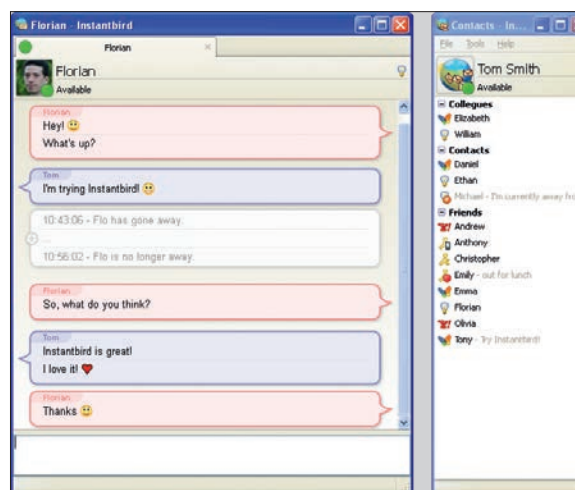
Усилия законодателей в области защиты копирайта заметно повысили популярность анонимной сети Tor как метода доступа к заблокированным ресурсам. Нетрудно догадаться, что рост востребованности подобной услуги инициировал появление других интересных проектов.

Разработчики Tor уже объявили о создании сервиса обмена сообщениями, обеспечивающего анонимность участников разговора для всех, кто не принимает в нём непосредственного участия и защищённого от слежки извне. Пользователю будет доступен некий пакет клиентского ПО *Tor Instant Messaging Bundle*, позволяющего делиться информацией через скрытые сервисы, не имеющие конкретного адреса и местоположения.

В основу *Tor Instant Messaging Bundle* легли технологии, созданные в рамках проекта Mozilla. Однако в качестве клиента будет использоваться не *Firefox*, а программа *Instantbird*.

Первые экспериментальные версии появятся в конце марта и будут весьма ограничены с функциональной точки зрения. В частности, в них будут отсутствовать возможности шифрования и аутентификации, которые будут внедрены позже по мере разработки.

Любопытно, что сначала на роль клиента претендовала программа *Pidgin*, но в конечном итоге выбор пал именно на *Instantbird*. В качестве основной причины такого решения называется недостаточное внимание разработчиков *Pidgin* к вопросам безопасности.



Программа Instantbird выбрана штатным клиентом защищённой сети Tor Instant Messaging Bundle.

ПРОЕКТ — СООБЩЕСТВУ

Persona передана сообществу

Mozilla объявила о прекращении самостоятельной разработки сервиса Persona.

Система идентификации пользователей Persona больше не будет развиваться в рамках проекта Mozilla — принято решение передать его под управление сообщества. Правда, подарком это назвать сложно.

Сервис разрабатывался как альтернатива аналогичным системам OpenID и OAuth. В качестве основы использовался протокол BrowserID, что позволяло регистрироваться на любом сайте без создания отдельной учетной записи, используя

вместо этого запись в единой базе данных, где ключевым идентификатором являлся адрес электронной почты.

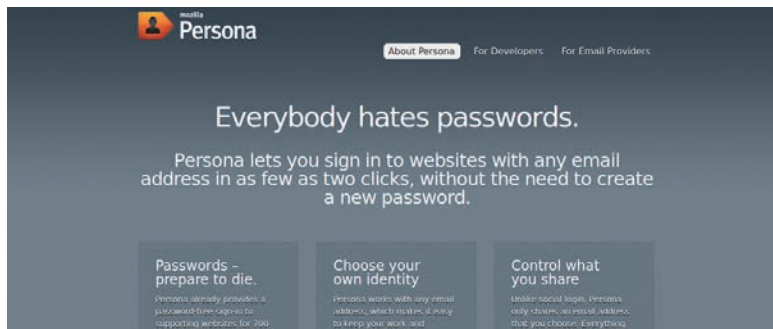
Persona поддерживала криптографию по открытым ключам и не отслеживала посещаемые пользователем сайты. Она всего-навсего подтверждала подлинность идентификатора, поэтому не создавала никаких угроз приватности. То есть, как ни взгляни — замечательная вещь.

Однако в качестве главной причины передачи проекта в руки сообщества

называется крайне низкая популярность решения. Владельцы сайтов предпочитают использовать другие системы идентификации, в том числе и такие, которые сложно отнести к системам идентификации как таковым — например, вход через учетную запись Facebook. В этой связи Mozilla была вынуждена признать, что функциональность Persona чрезмерно раздута, а поскольку базовое API уже фактически стабильно, то никакой особой работы для дальнейшего развития проекта не требуется.

Иными словами, сообществу передаётся практически готовый, но мало кому нужный продукт, нуждающийся разве что в активном маркетинге. Но именно этот род деятельности традиционно не был самым сильным у сообщества.

Не исключено, что главная причина отказа от дальнейшей разработки Persona носит имя Firefox Accounts — проект, имеющий для команды значительно более высокий приоритет. Кстати, именно к работе над этим проектом приступают высвободившиеся сотрудники.



► Проект Persona будет передан под управление сообщества.

GNOME 3

На альтернативные платформы

Разработчики Gnome не против портирования.

Критики по адресу разработчиков Gnome 3 звучит довольно много. Разумеется, подавляющее большинство замечаний относится к необычному интерфейсу, что вряд ли можно считать конструктивной критикой. Однако есть замечание, которое нельзя не признать серьёзным: оно касается затрудненности портирования Gnome на отличные от Linux системы.

На эту критику ответил участник Gnome Release Team Маттиас Класен [Matthias Clasen].

Он признал, что разработчики в первую очередь ориентируются на пользователей Linux. Именно по этой причине проект не будет отказываться от расширения поддержки специфических сервисов *systemd*, тем более, что значительная их часть основана на механизмах, которые уже реализованы в *Gnome-settings-daemon*.

Тем не менее, разработчики Gnome 3 вовсе не против портирования оболочки на другие платформы. В качестве доказательств приводятся примеры успешного сотрудничества с проектами FreeBSD и OpenBSD — на эти платформы Gnome уже портирован.

Для реализации поддержки альтернативных платформ требуется немного — наличие заинтересованных лиц. Причём заинтересованных настолько, что смогут самостоятельно организовать всю необходимую работу.

Разработчики Gnome, в свою очередь, готовы принимать все патчи и подготовить решения для замены сервисов *systemd* на альтернативных платформах. Но главную работу должны сделать именно заинтересованные лица, поскольку основной проект не располагает для этого необходимыми ресурсами.

Таким образом, Маттиас Класен достаточно прозрачно намекнул сообществу на то, что базовых принципов СПО никто ещё не отменял. Хочешь, чтобы стало лучше — сделай это сам.



► Разработчики Gnome готовы помочь энтузиастам в портировании оболочки на другие платформы.

Организатор:



Участие для представителей ВУЗов и медицинских учреждений – **бесплатно** по обязательной предварительной регистрации.
Количество мест ограничено.

Всероссийский форум «ПРИОРИТЕТЫ 2014: Информатизация образования и здравоохранения в России»

4 июня 2014 г.

Отель «Холидей Инн Лесная»
Москва, Лесная, ул., д. 15

ЦЕЛЬ ФОРУМА:

Создание независимой всероссийской площадки для открытого диалога между разработчиками ИТ- и телеком-решений и профессиональным отраслевым сообществом, а также создание «банка» идей для практического применения бюджетными и частными образовательными и медицинскими учреждениями на своих площадках.

В ПРОГРАММЕ ФОРУМА:

- Пленарное заседание по вопросам информатизации образования и здравоохранения в России.
- Практическая секция «Education & Training 2.0. - Барьеры и точки роста информатизации научно-образовательного комплекса в России».
- Практическая секция «Health & Care 2.0. - Барьеры и точки роста «электронного здравоохранения» в России».
- Торжественная церемония награждения лауреатов всероссийского конкурса «Лучшие 10 ИТ – проектов для госсектора. Образовательные и медицинские учреждения», в рамках которой наградами будут отмечены заказчики и исполнители лучших инновационных проектов.

Стратегический медиапартнер:

Событийный партнер:

Информационные партнеры:



HR-партнер:



Официальный бизнес-партнер:



FIREFOX OS

Новые телефоны

Выпущены новые телефоны, работающие на Firefox OS.

На Всемирном конгрессе мобильных технологий был представлен ряд мобильных устройств под управлением Firefox OS. Есть среди них и весьма дешёвые аппараты.

Компания ZTE предложила потребителям два новых смартфона с экранами в 3,5 и 4 дюйма. Оба они оснащены одинаковыми процессорами Qualcomm Snapdragon, работающих на частоте 1,2 ГГц. Знаменитая своими модемами Huawei показала новую версию уже известной модели Ascend Y300. Отличие — сама система Firefox OS. Предложить пользователю только частичную новинку — ход осторожный и прагматичный, консерваторов это порадует. Фирма Geeksphone представила реально революционную модель, которая так и названа — Revolution. Её особенность заключается в том, что на смартфоне установлено две системы — Firefox OS и Android. По умолчанию запускается вторая, но при помощи специального приложения пользователь сможет переключиться на Firefox OS. Размер экрана устройства —

почти пять дюймов, что автоматически переводит его в разряд «почти планшетов». И начинка соответствующая: процессор Intel Atom Z2560 1,6 ГГц, 1 ГБ ОЗУ и 4 ГБ флэш-памяти.

Представлены две эталонные модели планшетов на основе Firefox OS. Первый выпускает компания VIA — пользователь получит семидюймовый экран, 8 ГБ флэш-памяти и две камеры. За 10-дюймовый вариант отвечает фирма Foxconn, оснастившая устройство даже графическим акселератором PowerVR SGX544MP2.

Эталонная линейка напоминает аналоги на Android. Видимо, со временем 7 и 10 дюймов станут стандартами де-факто — что разумно с практической точки зрения.

А китайская компания Spreadtrum решила одарить поклонников бюджетных решений: смартфон от Spreadtrum будет стоить всего \$25. Техническими характеристиками он похвалиться не может, но выпуском более чем скромного аппарата Mozilla и Spreadtrum планируют оценить объём рынка недорогих решений.

КИТАЙ

Red Flag спущен

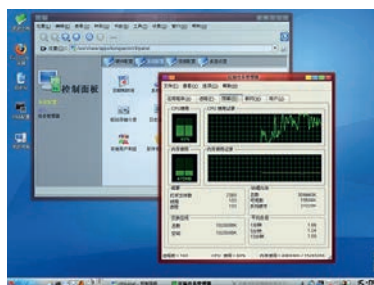
Проект Red Flag будет закрыт.

По сообщению издания South China Morning Post, проект создания китайской национальной ОС Red Flag в ближайшее время могут свернуть. У компании-разработчика нет денег, и её, вероятнее всего, придётся ликвидировать. По крайней мере, контракты с сотрудниками уже начинают расторгаться.

О причинах такого решения рассказывает сотрудник IDC в Пекине Эрик Пэн. Он считает, что проект так и не стал узнаваемым брендом и не выдержал конкуренции с Red Hat и SUSE. А для выравнивания шансов нужны слишком большие инвестиции. В общем, овчинка выделки не стоит.

Поражение в открытой и честной конкурентной борьбе дистрибутива Red Flag явно не грозило. Система создавалась исключительно для внутреннего китайского рынка. И даже не столько рынка, сколько госучреждений. Все прибыли и издержки прекрасно просчитывались

заранее. Ну, а в проникновение к западному пользователю вряд ли верили даже самые большие оптимисты. К тому же второе место по внедрениям внутри огромной страны — это отнюдь не провал. Подобными показателями в пору гордиться. Не исключено, что речь в данном случае идёт лишь о смене платформ — в Китае активно разрабатывается China Operating System, и страна вряд ли откажется от информационной независимости. **LXF**



➤ Китай прекращает разработку операционной системы Red Flag.

Новости короткой строкой

➤ Вышел релиз свободного гипервизора Xen 4.4, с функцией запуска гостевых систем в режиме EFI и улучшенной поддержкой архитектуры ARM. Источник: <http://blog.xen.org/index.php/2014/03/10/xen-4-4-released/>

➤ Бета-тестирование KDE 4.13 продлится до 16 апреля. Обновлены приложения и кое-что переделано для разработчиков. Источник: <http://dot.kde.org/2014/>

➤ В ответ на бурную критику со стороны поклонников консервативных интерфейсов в Ubuntu пообещали вернуть пользователям меню, размещаемое в окне приложения. Источник: <https://bregmatter.wordpress.com>

➤ Компания Google представила релиз web-браузера Chrome 33 с поддержкой спецификации Custom Elements. Это даёт возможность унаследования уже существующих элементов. Источник: <http://googlechromereleases.blogspot.ru>

➤ Компания Crytek портировала на Linux игровой движок CryEngine, в т.ч. для популярных игр *Crysis*, *Far Cry*, *Warface* и *Ryse: Son of Rome*. Источник: www.crytek.com

➤ Вышла стабильная версия лёгкого почтового клиента Sylpheed под номером 3.3.1, улучшившая поддержку популярного протокола IMAP. Источник: <http://sylpheed.sraoss.jp>

➤ Новая версия растрового графического редактора Krita из KDE улучшила поддержку планшетов. Источник: <http://krita.org>

➤ Вышла версия Mate 1.8. Из новинок — программа *mpaste*, для сохранения фрагментов текста на облаке. Источник: <http://mate-desktop.org>

➤ Приложение *Calibre*, предназначенное для организации личной библиотеки, с версии 1.26 можно установить в изолированном режиме, без прав root. Источник: <http://calibre-ebook.com>

➤ Вышла новая версия дружелюбного к новичкам дистрибутива Manjaro Linux. Источник: <http://manjaro.org>

➤ Облачная платформа SUSE Cloud 3 реализовала наработки и компоненты из OpenStack Havana. Источник: <http://www.eweek.com/cloud>

softline®



Services

Software

Cloud

ИТ-архитектура вашего бизнеса



РАЭК

КООРДИНАЦИОННЫЙ ЦЕНТР
НАЦИОНАЛЬНОГО ДОМЕНА
СЕТИ ИНТЕРНЕТ



Rutube



7 АПРЕЛЯ

ARENA MOSCOW

20 ЛЕТ

20.THERUNET.RU

ДОМЕНУ.RU

Весна. Интернет. Конференция.

**РИФ+КИБ
2014**

**23–25 апреля
пансионат «Поляны»**

www.rif.ru



Подарок для
подписчиков
журнала —
видео-трансляция
РИФ+КИБ 2013.

Для активации необходимо
перейти по QR-коду
и ввести промо-код:
RIF13VIDEO



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

Нет новостей? Отличная новость!

З а последнее время мы начинаем привыкать к тому, что на нас валятся новости. Нового в них частенько и нет: это просто хорошо испорченное старое. И ладно бы они портились сами — нет ведь, они еще и норовят испортить всё в округе...

И тем приятней иногда узнавать новости, в которых нет новоиспорченного. Например, о выходе... нет, не беты новой Ubuntu: доблестные зулусы отрешились от старого мира с его альфами и бетами. Но по всему Зулуленду объявлен приказ о заморозке релиза 14.04. Вы вправе сказать, что ничего нового тут нет? Да, не могу не согласиться. Более того, открою вам страшную военную тайну: и релиз был заморожен в стадии, когда в нём не было (почти) ничего нового. Не считать же таковым

Центр управления, отпочковавшийся от Gnome'ского — различий между ними не найти и под микроскопом. С другой стороны, радует, что в этом релизе умудрились не «слопать» настройку клавиатурных раскладок и их переключателей... Хотя нет, кое-что действительно новое есть: возможность включить локально-интегрированное меню как альтернативу глобальному. Это, в сочетании с тайлингом окон, одна из тех убойных мелочей, столь же приятных, сколь и полезных. И которые служат объектом иронии Ubuntu'фобов. После чего втихаря заимствуются ими.

И еще: возможно, это будет последняя Ubuntu в привычном нам виде. Но она будет гордо нести высокое звание долгоиграющего релиза. Чем поможет пережить трудное время.
alv@posix.ru

Сегодня мы рассматриваем:

Quirky 6.1.1 14

У вас завалился старый компьютер? Вместо того, чтобы собирать пыль, пусть поработает с этим ответвлением легковесного дистрибутива Puppy Linux — еще более легковесным!

Securstore Cloud 15

Мы решили взглянуть на этот сервис резервного копирования класса предприятия и убедиться, что им несложно пользоваться и он действительно обеспечивает сохранность наших данных.

Fritzing 0.8.5b 16

Электронщики-самоделкины, планируйте и создавайте собственные схемы с помощью этого популярного

открытого приложения. Может, изобретете новый Raspberry Pi...

phpMyAdmin 4.1.5 17

Свежая версия свободного инструмента администрирования MySQL произвела на нас отличное впечатление.

NAS4Free 18

С этим удобным дистрибутивом вы можете преобразовать купленное в магазине хранилище NAS или создать собственное на имеющемся оборудовании.

ioSafe 214 19

Скройте свои файлы от пожаров и наводнений в этом огнеупорном

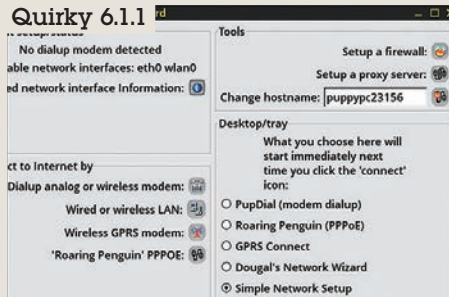


► **ioSafe 214:**
ваши данные —
как в швейцарском банке.

и водонепроницаемом хранилище NAS — оно как раз для этого.

To The Moon 20

Герой этой истории не склонен выть на Луну — уж лучше он на Луну отправится! Тем более, что это его давняя и пламенная страсть.



► К этому замечательному ответвлению приложил руку сам автор Puppy Linux.



► Попасть на Луну помогает несколько мрачноватый сервис выполнения последнего желания.

Сравнение: Пускачи приложений

с. 28

Gmrun



Homerun



Gnome-Do



Kupfer



Gnome-Pie



С этими удобными программами запуск вы мигом перепрыгнете в любимое приложение, не тратя драгоценных секунд. Они также улучшают внешний вид рабочего стола.

Quirky 6.1.1

Лес Паундер прогуливается с эксцентричным родичем Puppy Linux.

Вкратце

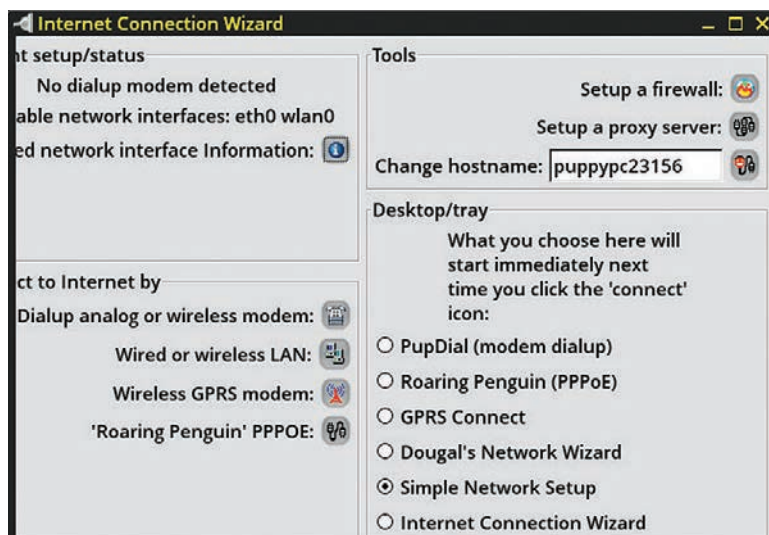
» Маленький и легкий дистрибутив для воскрешения старых компьютеров.

Технологии неуклонно совершенствуются, и за последние годы мы видели, как процессоры из одноплатных превратились в многоядерные, а их физические размеры значительно сократились. Со временем дистрибутивы Linux также выросли, и типичная установка весит теперь 4 ГБ. А что же со старой технологией? Мы выбрасываем ее на помойку или можем переработать в легкий Linux-дистрибутив?

Взглянем на Quirky [англ. странный], оригинальное ответвление Puppy Linux (отличный легкий дистрибутив с загрузкой 140 МБ), являющееся тестовым полигоном для новых идей; отсюда и имя Quirky. В 2009 году он сперва создавался как проект-игрушка Барри Каулера [Barry Kauler], создателя Puppy Linux, чьей целью было сделать дистрибутив как можно меньше.

Quirky верен идеалам Puppy: сделанный для устаревших ПК и оптимизированный для использования на USB-брелках и медиа-картах, он теперь весит 110 МБ. Если сравнить его с Puppy Linux, на первый взгляд видны лишь несколько косметических изменений. Настоящие отличия — под капотом, где Quirky произвел нестандартные изменения. Например, загруженный образ — это единственный раздел или накопитель, настроенный с использованием F2FS для гарантии мгновенного отклика на более старом оборудовании. Он также устанавливается целиком, заливаясь в память при загрузке initrd.gz временной корневой файловой системы.

На рабочем столе вы найдете версии с низким потреблением ресурса большинства общих приложений: например, LibreOffice заменен на Abiword и Gnumeric,



» Настройка Wi-Fi проходит легко благодаря предоставленному мастеру.

текстовый процессор и обработчик электронных таблиц. Рабочий стол умело представлен JWM (Joe's Window Manager): милый легковесный менеджер окон, который редко используется. Есть инструменты как для проводных, так и беспроводных интернет-соединений, поддерживается широкий выбор беспроводных микросхем. Но больше всего поражает, что Quirky в использовании проще Puppy — смелая заявка, если учесть, что Puppy прост донельзя.

Странный, но умный

Quirky может быть установлен на различные носители, и особенно подходит для таких, как USB- и SSD-накопители. Метод инсталляции требует некоторого знания о разделении и установке диска, но это не так уж сложно.

При использовании Quirky у нас была пара замечаний. Сначала не получилось действовать по инструкции, когда мы хотели копировать образ на USB-накопитель: все попытки провалились, в итоге пришлось использовать dd для копирования образа. Это не фатально, но некоторым пользователям может показаться колдобинной.

Второй вопрос касался настройки разрешения экрана, которая на исходном ПК Eee была 800×400. Ближайшее, что нам удалось получить без редактирования файлов конфигурации, было 640×480, но для тех, у кого стандартное разрешение, количество шагов может быть иным.

В любом случае, отличным свойством является возможность сохранения текущих настроек машины, так что при следующей загрузке не придется заново конфигурировать все железо и ПО.

Даже несмотря на свои замечания, мы продолжаем считать Quirky отличным. Он подойдет не каждому пользователю, но те, кому нужно старое оборудование с последней версией ядра, оценят такой достойный дистрибутив для своего инструмента. Выбор, сделанный Каулером, радикален, но окупается — получился быстрый и легкий современный дистрибутив. Можно утверждать, что потомство Puppy Linux доказывает его прекрасную родословную. **LXF**

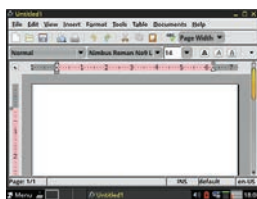


Свойства навскидку



Среда рабочего стола

Используя JWM, Quirky ухитряется быть многофункциональным, но легким.



Нужные приложения

Компактность дистрибутива не означает экономию на важных приложениях.

LINUX FORMAT Вердикт

Quirky 6.1.1

Разработчик: Барри Каулер
Сайт: <http://bkhome.org/quirky>
Лицензия: GPL и различные другие

Функциональность	8/10
Производительность	9/10
Удобство использования	9/10
Документация	8/10

» Маленький, экстравагантный, но идеальный для воскрешения старых ПК и нетбуков. Советуем дать ему шанс.

Рейтинг 9/10

Securstore 12.2

Месяц промаявшись с бесплатными решениями резервного копирования, **Маянк Шарма** тестирует Securstore, нацеленный на предприятия.

Вкратце

» Облачное решение для резервного копирования и восстановления, предлагающее шифрование и предназначенное пользователям на предприятиях малого и среднего бизнеса. См. также: *Unitrends* и *Avamar*.

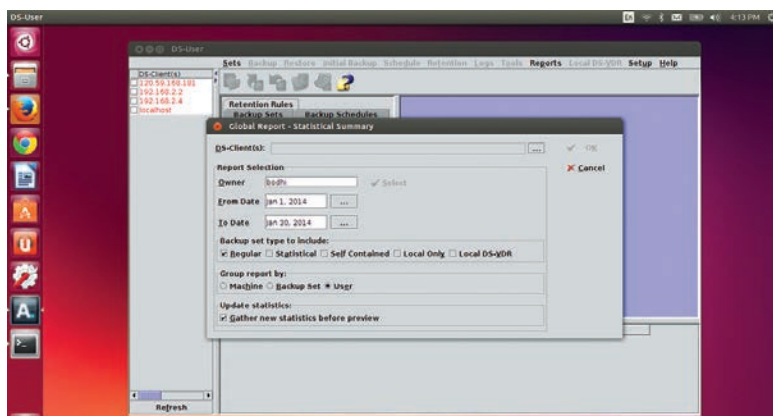
Обычно здесь рассматриваются облачные решения резервного копирования для среднего пользователя настольного Linux, а большинство из них рассчитаны только на одну машину. Но если вы отвечаете за ИТ-направление в среднем или крупном бизнесе или на предприятии, нужны более комплексные решения.

Сервиса Securstore Cloud Backup and Recovery предлагает именно это. Для обеспечения безопасности данных со всех многочисленных машин используется ПО облачного резервного копирования Asigra, затем данные переправляются в удаленные центры данных Securstore.

Уникальное свойство ПО Asigra — архитектура без агента: она не требует установки компонентов ПО на все машины, где нужно произвести резервное копирование. ПО использует стандартные API для захвата запрашиваемых данных внутри локальной сети, а также выходит прямо на компьютеры и приложения.

Для использования Securstore нужно установить два компонента. Утилита *DS-Client* делает всю черную работу: отвечает за резервное копирование и восстановление задач, а также связывает вашу сеть с центрами резервного копирования Securstore. *DS-User* — графический интерфейс пользователя, позволяющий создавать наборы резервного копирования.

DS-User требует наличия Java и прекрасно работает с OpenJDK. Официально компания поддерживает для *DS-Client* только серверные дистрибутивы от Red Hat и SUSE, мы выяснили, что и на Ubuntu



» Securstore — не для дома, так что не станем критиковать его интерфейс из 1990-х.

Server он также работает без проблем. Еще для *DS-Client* требуется сервер базы данных *PostgreSQL*. Установку ПО выполняет графический инсталлятор. По умолчанию он поставит *DS-User* и *DS-Client*, а если вы хотите управлять всеми наборами их резервных копий, сами установите *DS-User* на несколько машин в своей сети.

Комплексное решение

Создать набор резервных копий помогает мастер. Можно построить несколько различных типов, в зависимости от того, что конкретно вы сохраняете. ПО умеет сохранять данные локальных файловых систем, так же как и любых видимых в области локальной сети серверов, установленных с помощью сетевой файловой системы или безопасной оболочки. Каждый пункт резервного копирования по стандарту будет хранить 30 версий, но по требованию заказчика это число можно изменить. Возможно восстановить отдельный файл и определить, в какой именно из его версий. В помощь созданию графика резервного копирования на разных машинах Securstore позволяет присвоить приоритет, на выбор из 99 уровней. Можно делать резервную копию для множества машин, но компания рекомендует ограничиться все же семью при использовании 32-битного клиента и двенадцатью — для 64-битного.

DS-Client не хранит никаких данных, а действует просто как шлюз для их прохождения в сжатом состоянии, дедуплицированных и зашифрованных. Дедупликация — это процесс, гарантирующий, что конкретные данные не передадутся

на удаленные серверы резервного копирования Securstore более чем один раз. Это осуществляется путем создания цифровой подписи для каждого файла, прошедшего копирование, затем идет сравнение с более ранними резервными копиями. Для гарантии безопасности ваших данных Securstore использует сертифицированный модуль шифрования FIPS 140-2.

Люди из Securstore справились с заданием. Они используют высокоавторитетный бренд ПО резервного копирования и сочетают отличный сервис с круглосуточной поддержкой. Хотя настройка сервиса Securstore требует времени, а его кривая обучения чуть круче, чем у средних решений облачного резервного копирования, компания предлагает практическую помощь и подробную документацию для эффективного использования. **LXF**



Свойства навскидку



На базе Asigra

Securstore опирается на Asigra, увенчанного лаврами канадского профи.



Установка на практике

Представитель компании в деталях объяснит процесс установки.

LINUX FORMAT Вердикт

Securstore Cloud Backup

Разработчик: Securstore UK
Сайт: www.securstore.com
Цена: £500 + НДС/месяц/1 ТБ

Функциональность	9/10
Производительность	9/10
Удобство использования	5/10
Оправданность цены	7/10

» Securstore предназначен для предприятий, которым нужно комплексное хранение данных, а не для домашних пользователей.

Рейтинг 8/10

Fritzing 0.8.5b

Лес Паундер пробует популярную утилиту моделирования монтажных схем.

Вкратце

» Програма для создания прототипов и схем устройств пригодится радиолюбителям и им подобным...

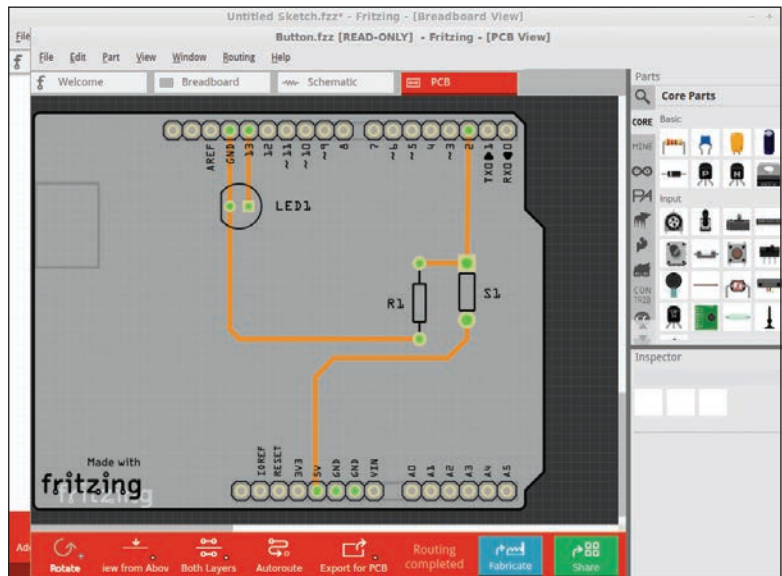
В настоящее время наблюдается всплеск интереса к разработке и созданию электронных приборов из серии «сделай сам», многие из которых основаны на устройствах типа Arduino и Raspberry Pi. Однако документирование этих проектов, особенно создание схем монтажа, из которых было бы видно, как компоненты соединены вместе, может быть довольно трудоемким. *Fritzing*, выходи! — вот наше спасение.

Fritzing — открытая программа, с помощью которой можно рисовать схемы высокого качества. Она обладает невероятно удобным интерфейсом, и каждый, кто знаком с графическим редактором *Inkscape*, почувствует себя как дома.

Для установки *Fritzing* достаточно загрузить архив с <http://fritzing.org> и распаковать его. В полученном каталоге вы найдете скрипт для запуска приложения. В большинстве свежих дистрибутивов Linux программа запустится без проблем, а если у вас дистрибутив постарше, убедитесь, что версия *libc* превышает 2.6.

С самого начала программа предлагает вам макет, на котором можно создать схему из компонентов и проводов. Чтобы нарисовать соединение проводом, достаточно щелкнуть левой кнопкой мыши на одном компоненте и дотянуть линию до другого.

Компоненты можно добавлять в проект из библиотек, называемых «ящиками [bin]». Они находятся в правом верхнем углу окна. Некоторые из самых популярных компонентов будут выбраны по умолчанию, но если среди них нет нужного вам, можно отфильтровать список, прокрутить



» С помощью *Fritzing* легко создать макет печатной платы для последующего изготовления.

его содержимое или воспользоваться поиском. В списке компонентов можно найти все, от резисторов до целого Raspberry Pi; кроме того, доступны распространенные схемы. Благодаря таким компаниям, как Adafruit, которые предоставили полные библиотеки компонентов для *Fritzing*, импортировать ящики из других источников впечатляюще просто.

Названия компонентов задаются в редакторе компонентов. Кроме того, *Fritzing* может автоматически проверить вашу схему и отобразить все ошибки. К компонентам можно добавлять примечания, с помощью которых удобно описывать новые идеи или делиться схемами с учебным классом.

вашу схему, а они изготовят по ней печатную плату.

Еще одна полезная возможность, польза которой проявляется не сразу — возможность выписать счет на используемые в проекте компоненты. Это очень удобно для желающих сэкономить.

Fritzing — замечательная программа, обладающая полезными возможностями, и простота ее интерфейса и набор функций улучшились по сравнению с предыдущими версиями. Если вы занимаетесь созданием электронных устройств, она позволит вам задокументировать свои схемы. **LXF**



Свойства навскидку



Масса компонентов

Огромный выбор компонентов благодаря библиотекам компонентов от крупных поставщиков.



Простота

Рабочий процесс во *Fritzing* паразитически похож на таковой в редакторе *Inkscape* или *Diagrams*.

Берем на борт

Fritzing также обладает прекрасными возможностями по созданию общетехнических чертежей, например, схем электрических цепей. Также на одной из вкладок можно просмотреть, как будет выглядеть готовый проект после переноса на печатную плату.

Чертеж печатной платы легко экспортировать в несколько доступных форматов, включая PNG или SVG, или распечатать его для последующего травления платы. Травление — это создание платы по вашей схеме, и в программе есть ссылки на компании, которым можно отправить

LINUX FORMAT Вердикт

Fritzing 0.8.5b

Разработчик: Interaction Design Lab
Сайт: <http://fritzing.org>
Лицензия: GPL V3

Функциональность	9/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» Простое в использовании приложение с массой функций, которое поможет самоделке получить печатные платы высокого качества для своих проектов.

Рейтинг 9/10

phpMyAdmin 4.1.5

Управлять вашим *MySQL* еще никогда не было так просто, и, как выяснил Дэвид Хейвард, становится проще с каждой новой версией.

Вкратце

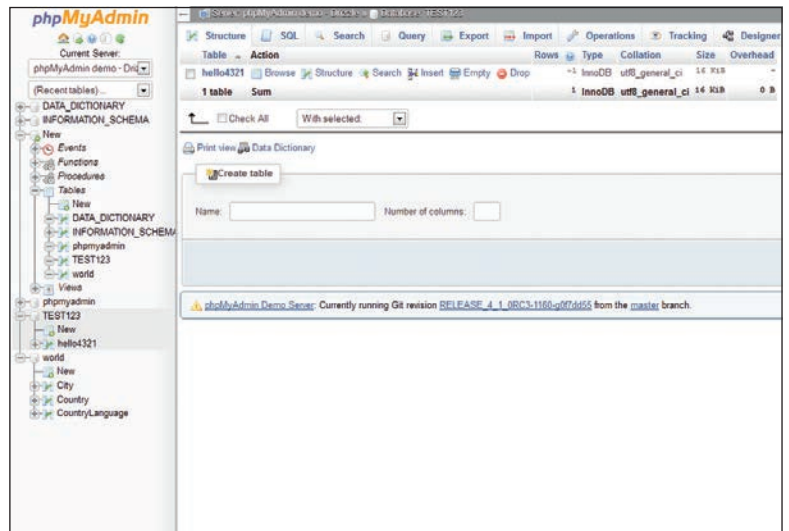
- » Web-сервер: Apache, IIS.
- » PHP 5.3.0 и выше, с поддержкой сеансов, Standard PHP Library (SPL) Extension, поддержка JSON, расширения PHP Zip и mbstring.
- » Поддержка баз *MySQL 5.5* и выше, *MariaDB 5.5* и выше или *Drizzle*.

Новейшая версия *phpMyAdmin* представляет еще более усовершенствованный графический интерфейс для управления базами *MySQL*. К счастью, переработка не затронула общий дизайн и удобную функциональность. Однако есть ряд важных изменений в других сферах, по сравнению с предыдущими версиями *phpMyAdmin*, и они определены к лучшему. Так, немного изменилось представление Списка таблиц [Table List]... но давайте обо всем по порядку.

Сборка 4.1.5 заявлена как версия с исправлением ошибок, кроме того, содержащая ряд нововведений. Во-первых, теперь можно прерывать открытие страниц, а значит, не нужно ждать, пока одна страница полностью загрузится, чтобы щелкнуть по следующей ссылке. Во-вторых, при создании новой базы данных появилась колонка Сопоставление не задано [Database Misses Collation]. В третьих, поле имени таблицы больше не пустует, когда вы щелкаете на Списке таблиц при ее использовании.

Кстати о Списке таблиц. Отображаемый на левой панели, теперь он обновляется, как только таблица получает новое имя. Вроде бы ничего особенного, но это избавляет от необходимости постоянно перезагружать весь список баз и таблиц после каждого переименования; хотя насколько часто вы это делаете, зависит от характера деятельности администратора или организации.

Также в этой версии исправлена ошибка с обновлением пользовательских



» Как только вы разберетесь с меню, управлять базой данных станет проще простого.

привилегий в подменю Обзор пользователей [User Overview]. Многие админы баз данных этого могут и не заметить, и тем не менее улучшается общее впечатление от работы с *SQL*.

Постоянное развитие

Ранее *phpMyAdmin* порой критиковали за использование Ajax — правда, злопыхатели оказывались в меньшинстве. Большинство администраторов баз данных и web-разработчиков были очень даже довольны скоростью, функциональностью и простотой работы с *phpMyAdmin*. Понятно, что предстоит еще много что доделать изнутри, в версиях от 4.1.6.0 до 4.2.0.0 будет ещё больше исправлений и обновлений, которые, будем надеяться, сделают этот инструмент администрирования ещё на шаг к ближе совершенству для его постоянных пользователей.

PhpMyAdmin уже давно стал незаменимым средством разработки для баз данных *MySQL*, и регулярные исправления ошибок — спасенье для тех, кому в работе дорога каждая минута. И хотя эффективности *phpMyAdmin* не занимать, всё же задумываешься: так ли нужна эта скрупулезность большинству пользователей? Мы бы предпочли, чтобы разработчики сосредоточились на потенциальных основных релизах, хотя, безусловно, отдаем им должное за столь кропотливый труд. Подобные обновления доказывают

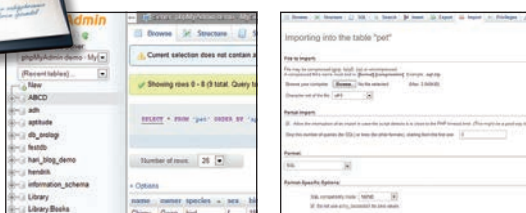


» *phpMyAdmin* — по-прежнему один из любимейших инструментов админов и разработчиков во всём мире.

их преданность своему делу и свидетельствуют об активности проекта *phpMyAdmin* и желании не останавливаться на достигнутом. **LXF**



Свойства навскидку



Web-интерфейс

Доступен и прост в использовании, и поддерживает большинство элементов *MySQL*.

Импортирование

Поддерживает аналоги *SQL*, *CSV* и частичный импорт более крупных файлов с опциями форматирования.

LINUX FORMAT Вердикт

phpMyAdmin

Разработчик: The VectorLinux team
Сайт: www.phpmyadmin.net
Документация: <http://bit.ly/1bEsd34>

Функциональность	9/10
Производительность	8/10
Удобство использования	8/10
Документация	9/10

» Релиз, направленный на решение определенных проблем, но поддерживающий множество функций *phpMyAdmin* в наилучшем состоянии.

Рейтинг 9/10

NAS4Free

Пуск в лучах славы купается NAS, зато NAS4Free подарит вашему старому ПК вторую жизнь в виде медиа-хранилища. Излагает **Дэвид Хейвард**.

Вкратце

» Ну очень впечатляющий дистрибутив NAS, на любой вкус, и постоянно улучшается.

NAS4Free — один из плеяды NAS-дистрибутивов Linux и Unix, полнофункциональных и исключительно неприязнительных в плане системных требований. Приятно, что установка и запуск не требуют особых усилий со стороны пользователя. Кроме того, в его основе — стабильнейшая FreeBSD, вкупе со множеством других технологий, превращающих его в один из любимых и лучших продуктов NAS из ныне доступных.

Новая версия 9.2.0.1.943, недавно выложенная на сайт проекта, содержит множество обновлений и исправлений, совершенствующих и без того замечательный продукт. Во-первых, сама база была обновлена до FreeBSD 9.2, теперь с поддержкой функций TRIM при использовании ZFS на твердотельных накопителях. Также в состав FreeBSD 9.2. входит обновлённая версия OpenSSL, 0.9.8y.

Слегка усовершенствован *WebGUI*: отныне он использует PHP 5.5. Это добавило ему быстродействия, без ущерба функциональности и удобства для чтения. Информация, которую *WebGUI* отображает, стала более детальной, появились дополнительные сведения в разделе статуса

```
Last login: Tue Jan 21 14:16:49 on ttyv0
Welcome to NAS4Free!
NAS4Free: 9.2.0.1 (revision 943)
Build on: Mon Jan 20 08:51:18 CET 2014 Platform: x64-full
Copyright (c) 2012-2013 The NAS4Free Project. All Rights Reserved.
Visit: http://www.nas4free.org for system updates!

LAN Network IPv4 Address: 192.168.1.250
LAN Network Interface: em0

Console Menu
-----
1) Configure Network Interfaces
2) Configure Network IP Address
3) Reset WebGUI Password
4) Reset to Factory Defaults
5) Ping Host
6) Shell
7) Reboot Server
8) Shutdown Server

Enter a number: █
```

» Основное меню, на первый взгляд, невелико, но не обманитесь — за этой простотой скрывается недюжинная сила.

[Status section] и улучшено визуальное представление меню.

В сборку интегрирована поддержка многочисленных контроллеров RAID, а в *WebGUI* — опция сжатия для ZFS lz4. Появилась поддержка ускоренного процессора A10-5700 от AMD, а также обновлены компоненты *Samba*, *gzip*, *PHP* и *Netatalk*.

Куча исправлений

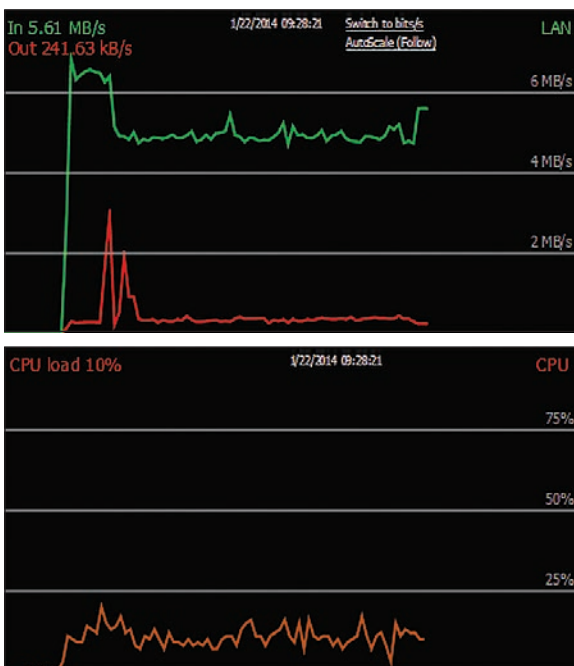
В предыдущей версии были неполадки с запуском DNS-ответчика. Эта, а также некоторые другие проблемы теперь решены. Кроме того, есть исправления ошибок в *WebGUI* и стабильности системы в целом.

NAS4Free представляет собой проект, который заметно вырос со времени появления первого кода FreeNAS, в 2011 году. Процесс полной установки на жёсткий диск достаточно прост, и, благодаря качеству установочного скрипта, по силам даже новичку. Однако большей популярностью у любителей NAS4Free пользуется встроенная [embedded] установка — на USB-носитель, которая ничуть не сложнее. На самом деле, весь процесс займёт всего несколько минут, в зависимости от того, сколько дисков с данными вы хотите включить и требуются ли вам какие-либо настройки RAID.

С улучшенным *WebGUI*, добавлять жесткие диски, создавать и редактировать разделы, создавать пользователей и группы и рассчитывать нагрузку на самопальный NAS-сервер стало очень просто. Если

вы на чём-то застряли или не можете разобраться с какой-либо функцией, в оживленном сообществе наверняка найдет кто-нибудь, кто будет рад вам с этим помочь.

Истинная прелесть NAS4Free в том, что вы можете начать с малого, а потом развернуться. Со временем можно увеличивать число ресурсов и расширять пространство, не углубляясь в саму структуру. Таким образом, эта система NAS подходит всем, от обычных пользователей до крупных организаций. Разработчики очень активно и непрерывно совершенствуют продукт, не создавая при этом сложностей для пользователей предыдущих версий. Что, в итоге, даёт NAS4Free сто очков вперёд, по сравнению с готовыми продуктами. **LXF**



» Новые элементы в разделе Статус NAS4Free необычайно информативны.

LINUX Вердикт
FORMAT

NAS4Free 9.2.0.1.943

Разработчик: Nas4Free
 Сайт: www.nas4free.org
 Лицензия: Упрощённая BSD

Функциональность	9/10
Производительность	9/10
Удобство использования	9/10
Документация	9/10

» Легко устанавливается, используется, поддерживается и обновляется. Великолепная производительность на надёжной основе FreeBSD.

Рейтинг 9/10

ioSafe 214

Спец по хранению данных **Нейл Мор** пытается свой NAS огнем и жидким льдом, типа водой.

Вкратце

» Устройство NAS с двумя дисками, способное уцелеть при пожаре и функционировать при полном погружении в соленую воду. Устройство повышенной устойчивости для реально тревожных и неловких.

Спецификация

- » Двухъядерный ARM-процессор Marvell Armada XP, 1 ГГц
- » 512 МБ ОЗУ
- » 2 диска SATA
- » Максимальная вместимость 8 ТБ
- » Возможность замены дисков непосредственно в процессе работы
- » Гигабитный LAN
- » 1×USB 2.0, 2×USB 3
- » 15×23×29 см
- » 12,7 кг

Конкуренция на рынке NAS еще никогда была такой свирепой. Падение цены и увеличение спроса на безопасные хранилища говорит о том, что, по крайней мере, NAS-устройства идут по пятаку за пучок. Учитывая, что FOSS обеспечивает мозги большинству накопителей NAS, понятно, почему они так быстро плодятся.

ioSafe 214 — последняя итерация двухсекционной модели NAS от ioSafe, которая не только предлагает избыточность данных, зеркалируя приводы, но также защищает от огня и воды. Это не просто слова: защита подкреплена сертификатом ASTM E199, поддерживает функционирование в течение 30 минут при температуре +843 °C, плюс способность до 72-часовой сохранности данных при погружении устройства в соленую воду на глубину порядка трех метров.

Мы пока не пробовали швырять ioSafe в огонь или с ним нырять, но видели его запеченным на барбекю и упавшим в воду; причем когда накопители вмонтировали в новый блок, их функциональность полностью сохранилась. Для критически важных локальных хранилищ, в нынешнем климате Великобритании, наводнение — реальный повод для беспокойства. Даже если попытаться решить проблему переносом устройства на верхние этажи, огонь всегда может уничтожить данные.

Не упущена из вида физическая защита. NAS поставляется с кенсингтонским замком и дополнительным, запирающимся на замок напольным шкафом, куда NAS полностью входит. Вне этого шкафа

на отсеке для установки дисководов замка нет, так что кто-то все же может отвернуть крышку и снять накопители, но ваши данные уже зашифрованы, так ведь?

Оборотной стороной столь всесторонней физической защиты является громоздкость. Размером с самый маленький 4-дисковый NAS или мини-сервер, весит он 12,7 кг — как шесть MacBook Pro. По-вашему, это нормально? И напоследок, но не в последнюю очередь, важно отметить: ioSafe настолько уверен в своем продукте, что добавляет к нему оплачиваемый сервис восстановления данных. Если ioSafe не сможет восстановить данные, то заплатит \$2500 за ТБ данных, потерянных в течение первого года, независимо от причины, по какой это приключилось.

По спецификации это 2-дисковый NAS, поддерживающий RAID 0, 1, 0/1 и JBOD. Доступ к дискам осуществляется через прикрученную переднюю панель, за которой прикручена еще одна панель, а внутри еще пара болтов, держащих шасси для вставки накопителя, на котором накопители надежно закреплены.

Настройка ioSafe 214 — левое дело, благодаря лицензированному Synology DSM. Это основанный на базе браузера интерфейс, функциональность которого не уступает прекрасному внешнему виду. Он обеспечивает Linux-подобную среду рабочего стола, GUI-доступ к различным вариантам настроек и любому контролю в части сервера.

С чего начинается поддержка, предлагаемая Synology DSM, понять сложно. Из поддержки здесь есть все для 2048 пользователей, iSCSI, внешний USB 3.0, загружающий ext3/4, FAT32 или NTFS, плюс еще ряд сервисов, включая обязательный доступ к облачному хранилищу. Для модуля видеонаблюдения выдается лицензия на одну камеру; поддерживаются обычный медиа-сервер DLNA/UPnP, iTunes, FTP, VPN и LDAP. Он исчерпывающий, простой в использовании, красивый с виду и расширяется с помощью хранилища приложений Synology.



» Тяжелый, крепкий, защищенный, черный, квадратный, устойчивый. Вот подходящие для ioSafe 214 прилагательные.

По части скорости ioSafe 214 достаточно резв, но не сногшибательно: приведенная производительность скорее соответствует 1-ТБ хранилищам в этой тестовой модели, а не самому NAS. Скорость длительной записи до 108 МБ/с неплоха, скорость разового доступа достигает 63 МБ/с, а асинхронные операции чтения — 80 МБ/с, тоже довольно мило. При такой цене, единственный Интернет-порт немного разочаровал, особенно принимая во внимание аварийноустойчивый тип винта, на который крепится эта модель. Мы еще отметили ограниченное число USB-портов — если добавлять принтеры и беспроводное электронное устройство защиты, придется озаботиться расширителями; но это уже мелкая придира. **LXF**



» Многослойные тепло- и влагоустойчивые системы обеспечивают безопасность данных.

LINUX FORMAT Вердикт

ioSafe 214

Разработчик: ioSafe 214
Сайт: www.ioSAFE.com
Цена: £549,99 (без дисков)

Функциональность	9/10
Производительность	8/10
Удобство использования	9/10
Оправданность цены	5/10

» Довольно дорогой, но приобретается за свои защитные в экстренных ситуациях характеристики, коих у него в избытке.

Рейтинг 9/10

To The Moon

Евгений Балдин реально растроган незатейливой историей любви...

Вкратце

» Интерактивная повесть. Неспешная, берющая душу история о странных людях.

Бесхитростная история любви. Кроме шуток. Не к партии, не к родине, а к Луне? Так ли уж Луна важна, да и вообще зачем она? Эта история придумана, сделана и озвучена одним человеком. Эта история зацепила множество людей. Почему?

Жил да поживал молодой канадский паренёк Кан Гао [Kan R. Gao]. Вместо того, чтобы как все (не все, конечно, но большинство), ходить на работу, смотреть хоккей по телевизору, пить пиво и быть довольным жизнью, Кан Гао делал игры. Что нужно для видеоигры? Наверное, сперва нужна идея. Визуальное представление также необходимо. Без звукового оформления тоже сейчас уже не обойтись. Ну и, наверное, нужно всё это заставить светиться и звучать в зависимости от выбора играющего. Иными словами, нужны автор, художник, композитор и немного программиста. Минимум четыре человека — всем нужно рассказать и показать, что именно требуется. За всеми нужно уследить, всех нужно как-то мотивировать — например, деньгами. А можно все эти профессии попытаться совместить в одном человеке по примеру Кан Гао. Он не смог заменить всех, но это действительно игра одного человека, в которой нет нестыковок и прорех. Если, конечно, закрыть глаза на жуткий игровой движок.

Кан Гао придумал историю, написал к ней музыку (можно послушать на сервисе Bandcamp), а в качестве движка использовал совершенно, казалось бы, не приспособленный для этой истории игровой движок RPG Maker XP. RPG Maker — это закрытое произведение сумрачного



» Луна и начало истории любви. Одна из немногих графических вставок за игру. Их Кан Гао не рисовал. Видимо, художественного дара в дополнение к дару музыкальному у него нет.

японского гения, предназначенное для выпечки домашних RPG японского же розлива. Длительное время даже переводы на другие языки, кроме японского, были исключительно нелегальными. Но лёгкость, которая позволяла полному профану в программировании набросать свою видеосказку, хоть и корявенькую, сделала RPG Maker довольно известным инструментом в среде энтузиастов-игроделов. Поддержка GNU/Linux официальными разработчиками и не планировалась, но энтузиасты — на то они и энтузиасты, чтобы делать то, что им хочется. Так родился проект mkxp (<https://github.com/Ancurio/mkxp>), который позволяет запустить готовые игровые скрипты и использовать ресурсы закрытой программы в свободном окружении. С помощью этого проекта *To The Moon* стал доступен другим энтузиастам, только энтузиастам GNU/Linux, в рамках Humble Indie Bundle X в самом начале 2014 года. Всё работает из коробки без проблем, если отвлечься от ужасного пользовательского интерфейса, «заточенного» под нужды любителей JRPG.

История довольно короткая. Её продолжительность — примерно четыре с половиной часа. Завязка фантастическая: некая фирма, с говорящим названием «Зигмунд», специализируется на фальсификации памяти. Замену памяти заказывают смертельно больные, чтобы «прожить» свою жизнь по-другому. Сделать то, что сделать не удалось, но очень хотелось, хоть и не на самом деле. А потом спокойно умереть. Два сертифицированных специалиста, Нил Уоттс и Ева Розалин, были

вызваны к Джону Уайлсу. Джон хочет слетать на Луну...

Ни в коем случае не читайте обзоров и «прохождений» перед игрой. Вся прелесть сосредоточена в истории. Необходимо привыкнуть к стилистике RPG Maker и насладиться игрой «актёров». В 2011 году, когда вышла эта история, она получила множество премий и номинаций за историю и за музыкальное сопровождение, а также самый высокий в том году рейтинг на Metacritic. В этой игре нет никаких RPG-боев!.. Ну-уу, почти.

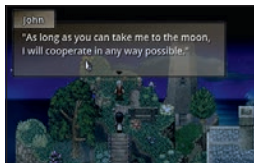
Русский язык официально не поддерживается (необходимо бегло читать на английском), но на форуме Freebird Games разделе Translation Projects есть тема для энтузиастов перевода на русский. В конце концов, энтузиасты делают то, что хотят — почему бы не сделать и перевод? **LXF**

Свойства навскидку



Запуск ракеты

Логичная история с достойной концовкой. Снабжена приятным звуковым сопровождением. И графика в общем симпатичная.



Переговоры

Пользовательский интерфейс — ужасное наследие RPG Maker XP. Элементы головоломки, похоже, введены для затягивания процесса.

LINUX FORMAT Вердикт

To The Moon

Разработчик: Freebird Games

Сайт: <http://treebirdgames.com>

Цена: 279 руб. на Steam; DRM-free-версия для Linux — рублей на 70 дороже.

Сюжет 9/10

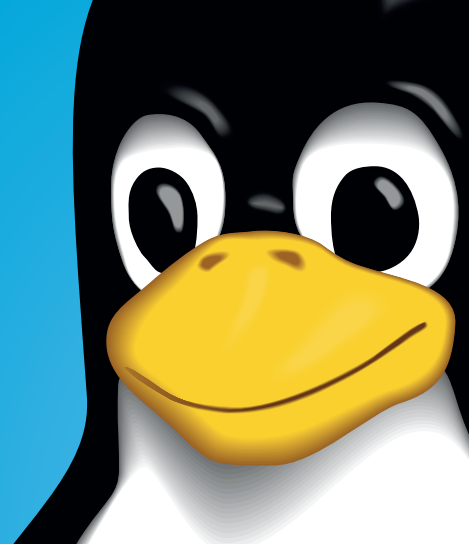
Графика 4/10

Увлекательность 8/10

Оправданность цены 6/10

» Милая, не затянутая история любви. Всё делалось ради неё, даже если вначале казалось иначе.

Рейтинг 7/10



Используйте свободное ПО — сэкономьте годовой бюджет!

Операционная система GNU/Linux поможет вам **с наименьшими затратами** решить проблему лицензирования программного обеспечения, навсегда избавиться от компьютерных вирусов и повысить надежность вашей компьютерной сети.



**С нашей
помощью
вы сможете**

**Сконцентрироваться
на своем бизнесе,**
не отвлекаясь на вопросы
поддержки своей
ИТ-инфраструктуры



**Забывать о вирусах,
угрозах безопасности**
и необходимости
лицензирования
программного обеспечения



**Оптимизировать
затраты**
на лицензирование ПО
за счет максимально
возможного использования
свободного ПО

ГНУ/Линуксцентр предлагает:

- Лицензирование во ФСТЭК.
- Внедрение наиболее дружественных вариантов ОС GNU/Linux и прикладных решений на базе свободного ПО.
- Абонентскую поддержку вашей сети.
- Обучение сотрудников вашей компании.

Наш опыт внедрения свободного программного обеспечения в организациях различного профиля поможет выбрать **оптимальное сочетание свободного и коммерческого программного обеспечения**, подходящее именно для вашей компании, а также поможет избежать технических и организационных проблем при внедрении свободного ПО.

Решите проблемы лицензирования ПО и поддержки компьютерной сети с помощью профессионалов!

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center



БРАУЗЕРЫ

Opera для Android

Обновлённая Opera поддерживает видеозвонки.

Несмотря на то, что скоро уже все сайты будут иметь собственные программы для чтения опубликованных материалов, браузер продолжает оставаться крайне востребованным приложением у пользователей средней степени продвинутой, т. е. у подавляющего большинства. Поэтому разработчики браузеров продолжают состязаться, предлагая пользователям новые возможности.

Компания Opera Software — не исключение. В новой версии браузера Opera для Android пользователи ждут обновление движка Chromium до 33-й версии, улучшенный дизайн экспресс-панели и поддержка HTML5 Vibration API. Действительно, на текущий момент Opera является одним из наиболее адаптированных к применению на небольших экранах браузеров.

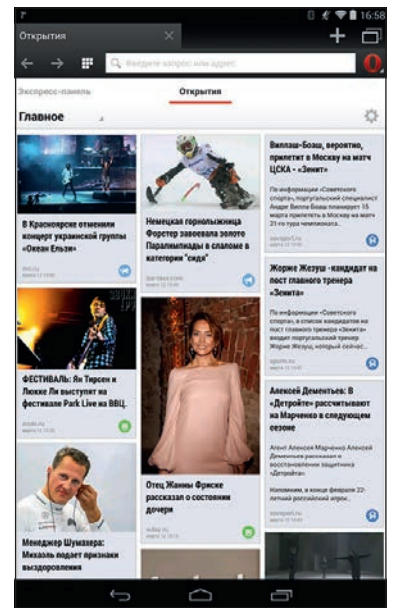
Однако главное новшество всё-таки в другом. В данной версии программа

Opera для Android поддерживает технологию WebRTC (Web with Real Time Communications), которая активно продвигается компанией Google в качестве основы для решения, позволяющего осуществлять видеозвонки и проводить видеоконференции без необходимости устанавливать дополнительное ПО.

Теперь пользователи браузера Opera смогут производить видеозвонки непосредственно из браузера. Главное — чтобы у собеседника тоже был установлен браузер с поддержкой WebRTC (не обязательно Opera). В общем, самая заметная разница с версией для настольного ПК ликвидирована.

Впрочем, пока рано говорить о востребованности этой функции именно в браузере Opera. Всё-таки в системе Android есть специализированные приложения для аналогичной цели.

» Браузер Opera продолжает оставаться одним из самых удобных для пользователей планшетов.



ПРОШИВКИ

Яндекс.Кит

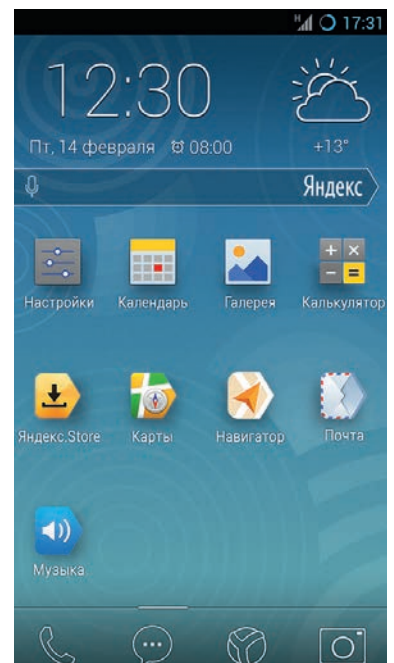
Представлена российская прошивка для смартфонов.

Разработчики российской компании «Яндекс» представили прошивку для смартфонов на базе ОС Android. Она предназначена для установки на новые аппараты при их производстве. Самостоятельная установка пользователем как штатная операция пока не предусмотрена. Очевидно, что главная цель прошивки — по умолчанию предоставить максимально полный доступ ко всем сервисам «Яндекса» — Яндекс.Навигатору, Яндекс.Поиску, Яндекс.Картам, Яндекс.Почте, Яндекс.Диску и т. д. Прошивка поддерживает единую регистрацию, а её основа — фирменный трёхмерный интерфейс.

Приятная новость для патриотов — приложения могут устанавливаться из Yandex.Store, и вовсе не обязательно привязывать устройство к учетной записи в Google. Увы, чрезмерно вольное обращение этой

корпорации с данными пользователей делает новую возможность серьёзным достоинством. «Яндекс» вряд ли многим лучше, но с ним, по крайней мере пока, не связано никаких скандалов. Ну, а что будет дальше — покажет только время. Владальцам смартфонов, работающих на Яндекс.Кит, бесплатно предоставляется 50-ГБ хранилище на облаке Яндекс.Диск. Это в пять раз больше, чем получают пользователи других систем на тех же условиях. Важное удобство — наличие у «Яндекса» своей платежной системы: пользователь может вообще не выходить за рамки фирменных сервисов и практически ничего от этого не потерять. Первые смартфоны, работающие на Яндекс.Кит — Huawei Honor 3 Yandex и Explay Flame. В ближайшее время они уже появятся в продаже. А летом появятся и планшеты с этой оболочкой.

» Весной этого года покупатели смогут приобрести смартфоны с прошивкой Яндекс.Кит.



БИЗНЕС

Android обрастает периферией

Компания Google приобретает Green Throttle Games.

Компания Google известна не только технологическими достижениями, но и весьма высокой бизнес-активностью. В частности, она очень любит покупать перспективные стартапы. Что, разумеется, не удивительно, если припомнить историю создания самой фирмы.

Конечно, значительная часть приобретений имеет весьма отдалённое отношение к системе Android. Однако покупка фирмы Green Throttle Games — совсем другое дело.

Компания Green Throttle Games занимается разработкой игровых манипуляторов для устройств, работающих на базе ОС Android. На первый взгляд — вещь абсолютно ненужная, поскольку смартфоны и планшеты используются в таких условиях, где сложно устроиться поудобнее. По крайней мере, вряд ли кто-то будет играть дома на планшете, а не на полноразмерном компьютере.

Особенно удивляет, что создаваемые Green Throttle Games манипуляторы позволяют множественное подключение к одному устройству. Представьте себе: пришли люди, например, на пляж, достали планшет и начинают вместе играть в какую-то стрелялку, отталкивая друг друга, чтобы увидеть происходящее на мониторе.

Но всё может оказаться значительно интереснее. Уже существуют мини-компьютеры, которые работают под управлением системы Android и подключаются к большому монитору. Как правило, они используются в качестве мультимедийных приставок.

Но почему они не могут быть игровыми консолями? Принципиальных препятствий для этого нет. Тем более, что на Android уже портировано значительное количество довольно интересных и сложных игр, а управление процессом при помощи естественного для планшета метода пока ещё далеко от совершенства. Ведь

не напрасно пользователям предлагаются виртуальные джойстики.

Пока неизвестно, понравится ли пользователям Android переучиваться с простых и очевидных схем управления игровым процессом на интуитивно непонятные кнопки. Впрочем, Google уже неоднократно доказывал, что умеет видеть смысл там, где его не видят другие.

► Не исключено, что такие микрокомпьютеры станут образцом для новых игровых консолей.



ПЛАНШЕТЫ

MediaPad X1

Очередной кандидат на самый лучший планшет.

Несмотря на изобилие предложений, современный рынок планшетов по сути довольно скуден. Моделей-то много, но большинство из них копируют друг друга.

Однако имеется ниша, в которой идёт нешуточная борьба. А именно — «самый-самый» планшет. Сделав такой пыталась и ASUS, и Samsung. Теперь в гонку вступила фирма Huawei. Их заявка на победу — планшет MediaPad X1. Даже название говорит само за себя: букву «X» где попало не ставят.

Толщина корпуса MediaPad X1—7,18 мм. Сравните с ASUS Nexus 7, который проигрывает по этому наиважнейшему параметру больше миллиметра, что по современным понятиям очень много. К тому же компания Huawei решила сделать корпус из алюминия.

Казалось бы, налицо проигрыш в весе. Но ничего подобного. MediaPad X1 весит 239 г, а ASUS Nexus 7—290 г. Это практически нокаут — и тоньше, и легче, и из металла, а значит, прочнее.

Для значительной части пользователей одного этого будет достаточно, чтобы расстаться с €399 — именно столько будет стоить новый планшет. Вероятнее всего, цена несколько завышена, если принимать во внимание только технические параметры. Однако при выборе устройств подобного класса на это мало кто обращает внимание.

Тем более, что начинка явно не хуже оболочки. Судите сами: четырехъядерный процессор Hisilicon Kirin 910 с тактовой частотой 1,6 ГГц, графический ускоритель Mali-450, 2 ГБ оперативной памяти и 16 ГБ встроенной.

Экран планшеты изготовлен по технологии IPS и имеет разрешение 1920×1200. Это гарантирует полный комфорт при выполнении любых пользовательских операций.

Разумеется, все ожидаемые интерфейсы тоже на месте — 2 камеры (13-мегапиксельная и 5-мегапиксельная), Wi-Fi, Bluetooth, 3G/LTE, слот для карт формата microSD и батарея емкостью 5000 мАч. На планшете будет установлена система Android 4.2 с оболочкой Emotion UI 2.0.

► MediaPad X1 — очередной кандидат на звание «самого-самого» планшета.





Охраним Android

Ваш смартфон — тот же ПК.
Маянк Шарма выясняет, как его обезопасить...

Вы знаете, если кто-то заинтересован в вашем точном местонахождении, оно выясняется в шесть секунд. Можно даже, вероятно, сделать вашу фотографию с журналом в руках и записать, как вы охнули. Не успеете вы и опомниться, как все ваши сообщения будут прочитаны, ваша контактная информация похищена, и отслежены не только ваша кредитная история, но и мастерство игры в *Angry Birds*.

И это не фантастика. У вас в кармане лежит лучший друг шпиона. Вы берете его с собой повсюду: от офиса до спальни, от улицы до уборной. Он записывает практически все, что вы делаете, и за считанные минуты его можно обратить против вас. Хотите — верьте, хотите — нет, но современный смартфон — самый жуткий кошмар для конфиденциальности частного лица.

Вспомните: ведь у вас там почтовые адреса и номера телефонов, планы, фото, а то и финансовая информация. Мало того, смартфон постоянно отслеживает ваше местонахождение, создавая подробнейшую карту ваших перемещений.

«97% приложений злоупотребляли доступом к частной информации.»

Есть куча способов, с помощью которых эти устройства рассылают информацию о своих владельцах, превращаясь в крупную проблему не только для желающих сохранить инкогнито, но и для простых обывателей. Даже если вы ни разу по своему смартфону не звонили,

вы все равно передаете на сторону информацию о себе — просто по факту обладания этим устройством.

Недавнее исследование, проведенное компанией HP, обнаружило, что 97% тестируемых приложений злоупотребляли доступом к частной информации, а еще 86% не имеют средств защиты от обычных взломов. Но хорошая новость в том, что ваши смартфоны позволяют всего за пару касаний изменить настройки конфиденциальности.

Давайте рассмотрим разные способы утечки информации с вашего смартфона — и минимизации подобной утечки. Также мы рассмотрим инструменты, позволяющие позаботиться о своей конфиденциальности, не затрагивая процесс обмена информацией.

За вами следят

Не позволяйте приложениям и сервисам надзирать за вами.



Многие задачи, некогда бывшие привилегией ПК, теперь переехали на телефоны: воспроизведение мультимедиа, запись, игры, GPS-навигация, и т.д. И для всего этого нужны приложения, а они — самое слабое звено в цепи между вашими частными данными и окружающим миром. Многие из них получают доступ к вашим персональным данным «для улучшения взаимодействия». И вы вынуждены верить, что данные употребляются только приемлемым для вас способом. Но увы, не все приложения четко определяют, что именно будет с данными, и способа убедиться в своей безопасности нет.

Далее, есть бесплатные web-сервисы. Web-компании вроде Google, Twitter, Facebook и т.п. предоставляют эти сервисы в обмен на информацию о вас, которую затем используют в целевой рекламе. Некоторые считают такой обмен справедливым, но те, кого волнует конфиденциальность, выражают все большую озабоченность.

Важнейший компонент вашего смартфона Android — система разрешений. При установке приложение сообщает вам, к чему ему нужен доступ. Затем можно или установить приложение, или нет. К сожалению, решение о том, являются ли эти требования доступа уместными, система возлагает на пользователя. А по результатам исследований (источник: <http://bit.ly/1bRbsVr>), запросы на доступ у многих приложений чрезмерны.

Есть ряд способов визуализировать разрешения приложений. Бесплатное приложение от Bit-Defender, *Clueful*, помогает вам узнать, что делает приложение и что оно должно делать. После

установки *Clueful* просканирует ваши приложения и распределит их по категориям Высокий риск [High Risk], Средний риск [Moderate Risk] и Низкий риск [Low Risk]. После этого вы можете просмотреть все списки, щелкая по приложениям, чтобы выяснить, к каким функциям у них есть доступ. Стоило бы удалить все приложения из категории High Risk, поскольку они могут отслеживать ваши пароли или читать сообщения электронной почты.

Есть еще мобильное приложение от Malwarebytes, *Anti-Malware*, которое включает также Privacy Manager. Оно сканирует приложения и делит их на категории в зависимости от доступных

либо функции, например, доступа к списку контактов, *XPrivacy* будет защищать реальные данные и вместо них предоставлять фальшивые контакты любому приложению, которое их запросит.

Кроме пресечения утечки информации в приложения, следует также минимизировать количество персональных данных, даже если вы всего лишь безобидно делитесь фотографиями. Джон Мак-Афи [John McAfee], скрывавшийся от властей, был засечен в Гватемале благодаря фото. Фотки многое рассказывают о вас благодаря прикрепленным к ним данным EXIF, и если вы снимали камерой со включенным GPS или смартфоном, это может выдать ваше местопребывание, время фотографирования и уникальный ID устройства.

Прежде чем поделиться изображениями, ликвидируйте их информацию EXIF

приложением *Instant EXIF Remover*. Интерфейса в нем нет: после установки оно будет доступно как опция в действии 'Share'. Будучи выбранным, приложение перехватывает все изображения, которыми вы хотите поделиться, и удаляет данные EXIF до передачи их в клиент электронной почты или другое приложение для раздачи доступа.

Перед загрузкой файлов на сервис облачного хранения, вроде Dropbox или Google Drive, неплохо было бы их зашифровать. Это легко сделать на компьютере с Linux с помощью *EncFS*. Инструмент *EncFS* доступен в репозиториях основных дистрибутивов, вроде Fedora и Ubuntu, и требует, чтобы вы создали две директории: одну для размещения незашифрованного контента, а другую — с зашифрованной версией. Работа *EncFS* заключается в том, что вы взаимодействуете с файлами в незашифрованных папках, и они на лету шифруются в зашифрованной папке. Чтобы использовать *EncFS* с облачным сервисом вроде Dropbox, просто поместите свою зашифрованную папку внутри папки Dropbox. Тогда любые изменения в зашифрованной папке автоматически будут синхронизироваться с Dropbox! Установив *EncFS*, создайте эти две папки с помощью `encfs ~/Dropbox/encrypted ~/Private`. Порасспросив вас, инструмент создаст папки. Отныне все файлы в директории **Private** будут синхронизированы.

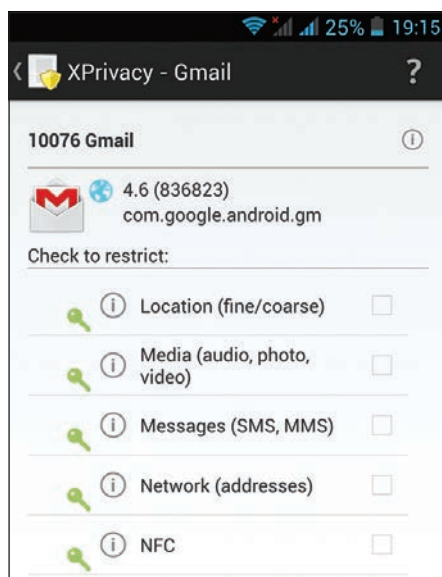
«Обмен изображениями многое рассказывает благодаря данным EXIF...»

им функций в телефоне: например, Access Calendar и Access Storage. Это очень удобно, когда вы, например, хотите просмотреть все приложения, умеющие считывать такую персональную информацию, как ваш список контактов или журнал.

Контроль за разрешениями

Определив приложение, нарушающее вашу конфиденциальность, можно его удалить. Недавно Google позволил некоторым функциям конфиденциальности просочиться в Android 4.3, и пользователи могут разблокировать их с помощью инструмента *Aps Ops* из Launcher, позволяющего выборочно отключать разрешения конфиденциальности: например, установить приложение распознавания музыки *Shazam*, но пресечь его способность отслеживать ваше местонахождение. Однако в следующем обновлении Google убрал эту функцию, к вящей досаде EFF. Отвечая на вопрос, Google сказал, что это функция экспериментальная и была выпущена случайно.

Если у вас устройство Android с правами root, вы можете получить эту функцию в виде модуля для среды *Xposed*. Пользователи таких устройств могут также использовать модуль *XPrivacy* для *Xposed*. Через *XPrivacy* контролируются конкретные разрешения для установленных приложений. Главная прелесть в том, что по отключении какой-



» Стоит овладеть нелегким искусством работы с *XPrivacy*, чтобы защитить свою конфиденциальность.

Троюродный террор

Вам не нужно общаться с подозреваемыми в терроризме, чтобы АНБ заинтересовалось вашими личными разговорами. Агентству разрешено «трехступенчатое» удаление от своей цели. Поэтому

они могут следить за людьми, разговаривающими с теми, которые разговаривают с теми, кто разговаривает с вами. Эти три степени удаленности позволяют АНБ накрыть практически кого угодно.

Общайтесь безопасно

Используйте свой телефон в режиме Incognito.

Ключ к защите телефона от любой слежки — одноуровневое шифрование. Все больше приложений и сервисов позволяют зашифровать данные на вашем устройстве перед их отправкой, а затем дешифровать их на устройстве получателя. Шифрование не защитит от кэширования данных, но предохранит данные от разглашения, сделав их нечитаемыми для всех, кто не обладает правильными ключами дешифрации.

Начните свою работу по блокировке данных с запутывания следов в Сети. Так же, как и в обычный браузер, в браузер Android тоже можно установить массу всяких дополнений. Популярны-

«Вы удивитесь, но многие по-прежнему используют смартфоны, чтобы звонить.»

ми дополнениями конфиденциальности являются дополнения *Phony*, для настройки пользовательского агента в браузере и сокрытия того факта, что вы используете мобильное устройство. Есть также дополнение самоуничтожающихся куки, которое удаляет все куки после закрытия сайта. Для более всестороннего контроля послужит дополнение *CleanQuit*, которое удаляет всю информацию о предыдущей сессии, включая журнал просмотров и скачиваний и закладки. Если вас заботит анонимность, стоит перейти на браузер *Orweb* (<http://bit.ly/1eiYkti>), преднастроенный на анонимный просмотр сети. Он также обладает плагинами для маскировки вашего устройства, управляет куки и не позволяет загружать контент *Flash*, и не ведет журнал. Ему нужен плагин *Orbot*, а *Orbot* — тот же *Tor* для Android (подробную информацию о проекте *Tor* см. на стр. 54). После первой загрузки *Orbot* запускает мастер быстрой

настройки. Если вы получили права root на своем телефоне, можете включить прозрачный прокси, который позволит всем сетевым приложениям автоматически работать через сеть *Tor*. Чтобы подписать и зашифровать сообщения электронной почты на своем мобильном устройстве, пригодится приложение *Android Privacy Guard (APG)*, являющееся реализацией с открытым кодом OpenPGP. Вам также понадобится приложение электронной почты *K-9*, отлично интегрируемое с *APG*.

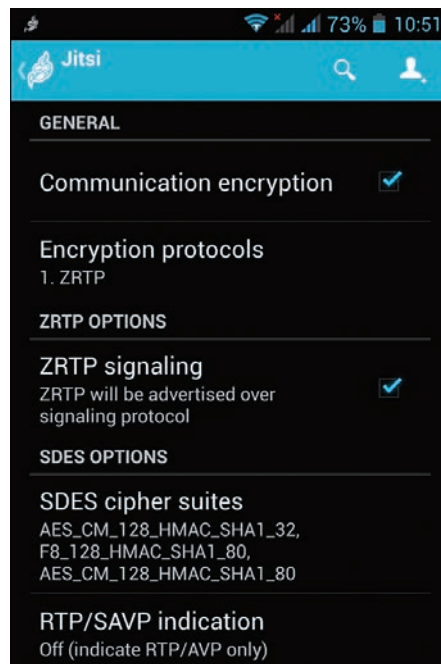
Для использования этих приложений сначала запустите *K-9* и настройте его на соединение с вашим сервером электронной почты. Затем запустите *APG* и нажмите на кнопку меню — выведется опция управления приватными и публичными ключами. Можете экспортировать эти ключи с настольного ПК и импортировать их в *APG*. После импорта *K-9* отобразит опцию подписи и шифрования, когда вы будете создавать новое сообщение электронной почты. Оно же потом и дешифрует новые зашифрованные сообщения, полученные по электронной почте.

Аналогично, если вы хотите шифровать быстрые сообщения, вам нужно приложение с открытым кодом *ChatSecure*. Это приложение использует протокол OTR для обеспечения безопасных сессий чата на учетных записях XMPP. С помощью этого приложения вы получите безопасный чат со своими друзьями в популярных сетях, включая Google Talk и Facebook, на любом OTR-совместимом клиенте, в том числе *Pidgin*, *Adium* и *Jitsi*.

Старая школа общения

Другая знаменитая форма мобильного общения — SMS. Сейчас популярность SMS пошла на убыль из-за спада цен на мобильный интернет, но многие по-прежнему считают текст основным средством общения. Сообщения SMS можно шифровать приложением с открытым кодом *TextSecure*. Но у получателя оно тоже должно быть, иначе ему будет не прочесть «шифровки». При первом запуске *TextSecure* позволит вам зашифровать все локально хранящиеся SMS, не тронув их исходные нешифрованные версии; рекомендуется удалить их после создания зашифрованных.

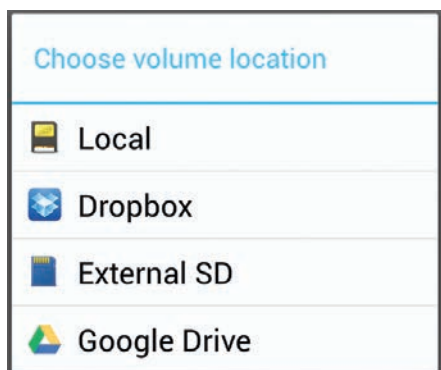
Прежде чем отправлять сообщения, надо создать безопасное соединение с устройством получателя, обменявшись ключами. *TextSecure* отправит получателю сообщение, на которое его приложение *TextSecure* автоматически ответит сообщением для установки безопасного соединения. С этого момента вы будете отправлять и получать зашифрованные сообщения.



► Можете использовать приложение Jitsi для зашифрованных видеозвонков.

Вы удивитесь, но многие из нас по-прежнему используют смартфоны, чтобы звонить, и есть приложения и сервисы, которые создают безопасные каналы, пресекающие все попытки перехватить ваши разговоры. Для зашифрованных звонков через Интернет вы можете использовать свободное приложение *RedPhone*. Есть также *SilentPhone*, разработанное Филом Циммерманом [Phil Zimmerman], который дал нам OpenPGP для обеспечения безопасности электронной почты и протокол ZRTP для обеспечения безопасности звонков VoIP. Приложение *SilentPhone* работает на нескольких мобильных платформах, но требует регистрационного взноса в размере \$10. Оба этих решения работают как обещано, шифруя звонки. Но их основное ограничение в том, что другой человек на линии должен иметь то же самое приложение.

Проект *Ostel* работает над этой проблемой. Они создали стандарт под названием Open Source Telephony Network (OSTN), на основе свободных протоколов и протоколов с открытым кодом, для одноуровневого шифрования голосовых каналов связи. Лучшее здесь то, что вы сможете связаться с любым пользователем, который использует приложение с поддержкой стандарта OSTN. Есть приложение *CSipSimple* для Android, *Acrobats* для пользователей iPhone, *PrivateGSM* для пользователей BlackBerry и кросс-платформенное настольное приложение *Jitsi* для пользователей Linux, Windows и Mac.



► Приложение *EncDroid* для Android позволяет создавать зашифрованные в *EncFS* папки и синхронизировать их.



Обезопасимся

Задраим порты и люки.

Помимо ограничения потока данных к сторонним приложениям и шифрования всех форм коммуникации, также надо защититься от физического вмешательства. Обеспечение безопасности устройства избавит его от самых частых угроз — кражи и несанкционированного доступа. Пользователи, озабоченные сохранением своей конфиденциальности, должны включить и применять одну из нескольких опций для блокирования телефона. Можно ограничить доступ через комбинацию, цифровой PIN, буквенно-цифровой пароль или голосовую команду. Если вы используете блокировку экрана, следует также отключить все виджеты блокировки экрана. Это требует наличия сторонних приложений, например, *Lockscreen Policy*, но сейчас они встроены в самую последнюю версию Android.

Блокировка телефона — дело хорошее, но она не спасет, если разблокированный телефон кому-то передадите. Прежде чем это делать, вы можете использовать *Screen Locker* (<http://bit.ly/LRBttz>) для блокировки экрана. Это приложение

отключает все формы ввода информации и не позволяет просматривать ничего, кроме того, что находится на экране. Для разблокировки устройства введите заранее настроенную комбинацию.

Есть приложения, блокирующие доступ к приложению с помощью пароля. Одно из них — *Privacy Master Free*: оно имитирует сбой, не давая приложению запуститься, и запрещает доступ к таким важным областям, как, например, Google

«Privacy Master Free имитирует сбой, не давая приложению запуститься.»

Play Store. Вы можете заодно блокировать менеджер задач и соединения USB. Затем есть приложение *AppLock*, которое, помимо способности блокировать доступ к приложениям, имеет два дополнительных хранилища, где можно прятать фото и видео. Приложение также может мешать переключению настроек типа Wi-Fi. Одна из его лучших функций — умение создавать профили блокировки. То есть можно создать список приложений, которые вы хотите заблокировать, пока вы в офисе, и еще один список — когда вы со своими шkodливыми племянниками. Блокировку можно включать в зависимости от времени и места. С точки зрения безопасности, приложение может случайным образом реорганизовать свою цифровую клавиатуру, чтобы посторонние не прознали ваш пароль, следя за движениями ваших пальцев. Также *AppLock* позволяет скрывать приложение из набора, чтобы его присутствие на вашем устройстве оставалось тайной.

Зашифруйте устройство

И, наконец, обеспокоенные сохранением конфиденциальности пользователи должны шифровать данные на своем телефоне с помощью встроенной функции. Однако в этом процессе есть ряд оговорок. Во-первых, шифрование — процесс односторонний: если вы его включили, у вас не будет

механизма выключения, или придется сбросить настройки телефона в заводские и потерять все свои данные. Также позаботьтесь сделать резервные копии своих данных перед началом процесса шифрования, и не прерывайте этот процесс. Прервав его, вы уж точно потеряете данные и приведете устройство в негодность. Прежде чем приступить, убедитесь, что вы задали пароль или PIN блокировки экрана, поскольку Android будет использовать его в качестве вашего ключа

дешифровки. Чтобы запустить шифрование, перейдите в *System Settings > Security > Encrypt device* [Параметры системы > Безопасность > Зашифровать устройство]. По завершении нужно будет вводить PIN или пароль при каждой загрузке телефона.

Вы можете решить зашифровать не все устройство, а только отдельные файлы. Одно из лучших приложений для этой цели — *SSE Universal Encryption*. Это приложение предлагает все популярные алгоритмы шифрования, включая AES-256, Serpent-256 и Blowfish-256, и имеет три модуля. Модуль *Password Vault* позволяет безопасно хранить пароли и размещать их по папкам, модуль *Message Encryptor* шифрует фрагменты текста, но самая интересная опция — модуль *File/Dir Encryptor*. Он позволяет выбрать файл через встроенный браузер файлов и затем зашифровать его.

Установка *CyanogenMod* требует усилий, но если у вас получится найти компьютер Windows, вы сэкономите свои силы благодаря *CyanogenMod Installer* с <http://get.cm>. Энтузиасты свободного ПО могут пожелать использовать дистрибутив *Replicant*. Он основан на *CyanogenMod* и замещает все проприетарные компоненты Android их альтернативами с открытым кодом. *Silent Circle* Фила Циммермана связано с испанским производителем телефонов *Geekphone*, и для особо озабоченных проблемами конфиденциальности скоро выпустят телефон *Blackphone*, который будет работать на дистрибутиве *PrivatOS*. **LXF**

Установка *CyanogenMod* требует усилий, но если у вас получится найти компьютер Windows, вы сэкономите свои силы благодаря *CyanogenMod Installer* с <http://get.cm>. Энтузиасты свободного ПО могут пожелать использовать дистрибутив *Replicant*. Он основан на *CyanogenMod* и замещает все проприетарные компоненты Android их альтернативами с открытым кодом. *Silent Circle* Фила Циммермана связано с испанским производителем телефонов *Geekphone*, и для особо озабоченных проблемами конфиденциальности скоро выпустят телефон *Blackphone*, который будет работать на дистрибутиве *PrivatOS*. **LXF**



► Помимо шифрования, *SSE* также умеет безопасно удалять файлы, чтобы устранить всякую угрозу.

Перейдите на сторонние прошивки

Каждый год Google предлагает для скачивания базовую операционную систему Android, широко известную как AOSP. Многие разработчики берут эту версию и работают с ней, создавая собственную версию Android.

CyanogenMod — один из таких дистрибутивов Android, и один из самых популярных, с миллионными пользователями. Одной из причин его попу-

лярности является предоставляемый им полный контроль над вашим устройством и свобода от всех связей с Google, вашей сетью или производителем телефона. Стоит отметить, что команда *CyanogenMod* быстро выпускает заплатки к дырам в системе безопасности и исправляет ошибки, которые Google ликвидирует только при очередном релизе Android. Сторонние прошивки включают и приложение

Privacy Guard, которое дает вам больший контроль над приложениями и их разрешениями.

Более новые версии приложения также включают функцию *AppOps*, отредактированную Google в Android 4.3. С ней можно запретить отдельным приложениям получать доступ к вашим данным. Последняя версия *CyanogenMod* включает безопасное SMS-приложение *TextSecure* прямо в прошивке.

Сравнение

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

Пускачи приложений

Шашанк Шарма постоянно ищет новых друзей в Facebook/Twitter и более эффективные способы работы, и нарыл пять программ запуска приложений, способных, по его мнению, облегчить вам жизнь.



Про наш тест...

В отличие от своих старых воплощений, более новые программы запуска приложений вовсе не требовательны к ресурсам. И это может стать одной из причин, по которой вы решите дать им шанс. Целью нашего Сравнения было найти инструменты, не привязанные к конкретному дистрибутиву или рабочему столу.

Все инструменты в нашем списке, кроме *Homerun* — приложения KDE, могут работать в любой среде рабочего стола. Вам по-прежнему будут нужны библиотеки, чтобы это обеспечить, но программные репозитории вашего дистрибутива вам помогут.

Поскольку эти инструменты делают намного больше простого запуска приложений, счету в Сравнении им добавляет наличие солидной документации. Не все дистрибутивы предлагают в своих репозиториях самые свежие версии, и мы сравним также процесс установки. А еще — протестируем, насколько поддаются настройке их внешний вид и клавиши быстрого запуска.

Наша подборка

- » Gmrun
- » Gnome-Do
- » Gnome-Pie
- » Homerun
- » Kupfer

Программы запуска приложений [launcher] прорвались на сцену разработки настольных систем несколько лет назад, и с тех пор обзавелись солидным количеством страстных приверженцев, одержимых идеей экономии времени. Если вы никогда еще такими не пользовались, не беда: они не успели завоевать всеобщее внимание. Но если вы недавно использовали Unity в Ubuntu или Gnome Shell, то уже сталкивались с функциями, изначально представленными в этих программах запуска приложений.

Цель программ запуска — помогать быстро и эффективно отслеживать приложения и файлы на вашем компьютере. Идея заключается в выдаче вам доступа к искомому нажатием на пару клавиш, вместо длительного просеивания иерархии меню в поисках нужного приложения.

Со временем такие программы обросли разными плагинами, которые, например, могут запускать команды в терминале, искать нужных людей в списке контактов и автоматически загружать фотографии. И со временем — включили поиск по рабочему столу; а значит, их также

можно использовать для поиска файлов и папок без всей этой возни с меню и директориями в файловом менеджере. Теперь вы найдете эти функции практически во всех современных дистрибутивах, независимо от вашего рабочего стола — Gnome, Unity, KDE или чего-то капитально иного.

В нашем Сравнении мы обратимся к старой школе. Мы отобрали пять инструментов, чутких к потреблению ресурсов и богатых функциями: от реверсивного поиска до автозаполнения, эти инструменты умеют делать все.

Функции и инструменты

Что такого особенного в программах запуска?

Программы запуска приложений служат для повышения эффективности вашей работы. Они делают это, адаптируясь под ваши потребности и предлагая результаты на основании ранее вами использованного. *Homerun*, единственное приложение KDE в нашем списке, входит в Plasma, но не устанавливается по умолчанию в большинстве дистрибутивов KDE. Приложение предоставляет те же функции, что и Unity в Ubuntu. Вы можете использовать его для поиска файлов на своем рабочем столе или для запуска приложений. Можете также создать список любимых приложений и директорий, и даже перемещаться по папкам, не переключаясь в менеджер файлов.

Gmrun не предлагает функции поиска по рабочему столу прямо сразу. Однако его можно легко настроить для работы с *Catfish*, чтобы обеспечить эту функцию. Он не позволяет создавать списка любимых директорий, но вы можете отредактировать файл настройки для создания привязки клавиш к вашим любимым приложениям. Мы подробно рассмотрим обе эти функции при обсуждении плагинов.

Gnome-Do и *Kupfer* похожи и внешне, и в плане предлагаемых функций. Дело в том, что источником вдохновения обоим послужил *Quicksilver* для Mac OS X. Вы можете использовать оба для запуска приложений и поиска файлов. Но вы сможете это сделать не раньше, чем укажете директории, которые эти приложения должны индексировать и каталогизировать.

В *Kupfer* можно создавать закладки, и он тщательно отслеживает вашу работу. Со временем он выучит, что вам нужно, и будет отображать часто используемые объекты в начале списка. Например, если вы постоянно вводите **LXF** и открываете папку, где храните PDF журнала, то через пару раз он представит вам эту папку в начале списка поиска, как только вы начнете вводить **LXF**.

С помощью *Kupfer* каждый выбранный пункт будет показывать вам серию действий, например, Open, Copy, Move, Delete и Archive, в зависимости от выбранного приложения или файла. Вы можете



» Вы должны определить директории для индексирования как в *Kupfer*, так и в *Gnome-Do*.

выбрать несколько файлов, нажав на пятую после каждого выбора, и все подобные выборы группируются вместе. Когда вы, наконец, выбираете действие, оно немедленно применится ко всем выбранным файлам в группе.

В отличие от других инструментов, *Gnome-Pie* не предлагает функции поиска по рабочему столу. Еще один отказ от традиций в том, что он не предназначен для работы с клавиатурой. Это единственный инструмент в нашем списке, который требует использования мыши для запуска приложений.

Вердикт

- Gnome-Do ★★★★★
- Kupfer ★★★★★
- Gmrun ★★★★★
- Homerun ★★★★★
- Gnome-Pie ★★★★★

» В отличие от *Kupfer* и *Gnome-Do*, *Gmrun* заставит вас попутать с файлами настройки.

Установка и настройка

Является ли процесс дружелюбным к пользователю?

Все проекты предоставляют исходные tar-архивы, а также бинарники для дистрибутивов. Многие также предоставляют PPA для Ubuntu и его производных, как *Homerun*, а Fedora, Когога и большинство других дистрибутивов предлагают *Homerun* в своих

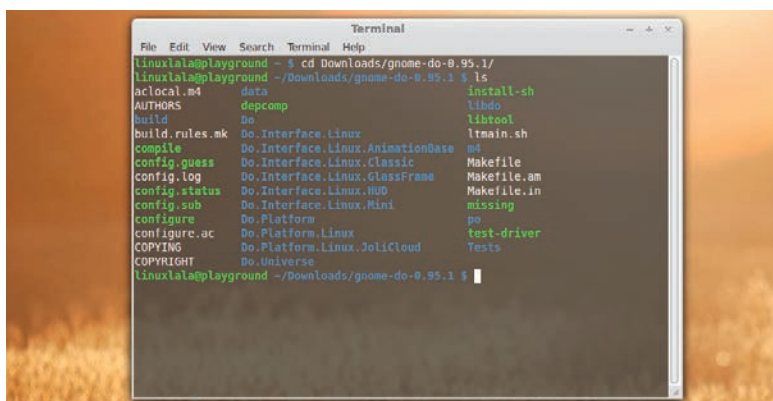
репозиториях. Поскольку это приложение KDE Plasma, вам придется установить полный рабочий стол KDE, если вы работаете не в дистрибутиве KDE.

Kupfer, *Gnome-Pie* и *Homerun* уже некоторое время не выпускали новых релизов. Это означает, что их можно без проблем

установить через инструмент управления пакетами своего дистрибутива. Также они предлагают исходные tar-архивы, если вы решите сделать установку вручную.

Gnome-Do после долгого периода бездействия недавно выпустил новый релиз, однако у него нет бинарников, а репозитории каждого дистрибутива по-прежнему предлагают старую версию, так что единственная опция — скомпилировать его вручную. Это потребует немало времени, поскольку проект Mono требует слишком большого количества библиотек и будет часто жаловаться на отсутствующие зависимости. К счастью, вы сможете установить отсутствующие с помощью менеджера пакетов своего дистрибутива.

Множество функций в *Gnome-Do* и *Kupfer* предоставляются в виде плагинов. Однако, в отличие от *Kupfer*, который предлагает плагины как часть комплекта, для *Gnome-Do* вам придется установить отдельный пакет плагинов.



» Tar-архив *Gnome-Do* не включает файла *Installation* или *Readme*. Вы должны запустить `./configure`, найти все зависимости и затем скомандовать `make` и `make install`.

Вердикт

- Gnome-Pie ★★★★★
- Gmrun ★★★★★
- Homerun ★★★★★
- Kupfer ★★★★★
- Gnome-Do ★★★★★

» *Gnome-Do* в арьергарде, поскольку установка слишком долгая.

Пользовательский интерфейс

Миловидный и приятный в использовании.

Поскольку программы запуска приложений созданы как замена существующим меню или программам запуска в среде вашего рабочего стола, интерфейс должен быть элегантным и интуитивным, или он совершенно не отвечает своей цели. Интерфейс должен давать пользователям возможность быстро

перемещаться от установленных приложений к местным файлам на монтированном устройстве хранения и до запуска браузера на требуемом сайте.

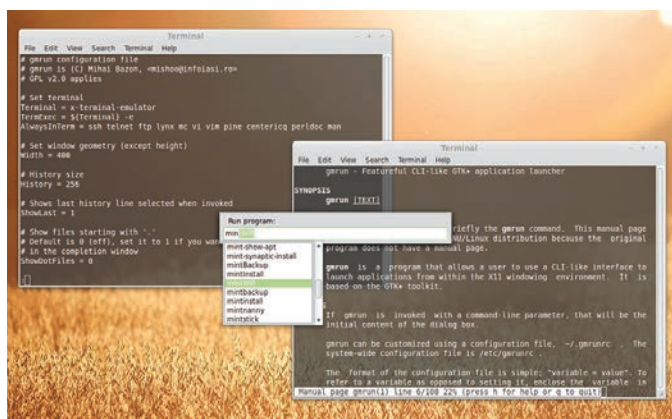
Программа запуска должна позволять создавать избранное, давая ему приоритет для отображения начале списка. Почти все программы

неплохо справились с этой задачей и моментально выдают свои предложения в процессе набора вами текста. Лучшим вариантом программы запуска будет программа не слишком прожорливая в плане потребления ресурсов на службе нашим целям и не путающаяся под ногами, пока в ней нет необходимости.

Gmrun ★★★★★

Интерфейс *Gmrun* — самый неприязнательный из всех инструментов в нашем Сравнении. Его интерфейс говорит: «Запустите программу [Run program]» и предоставляет текстовое поле, где вводится имя приложения, которое вы хотите запустить. В интерфейсе вообще нет значков, поэтому он не показывает, распознал он команду или нет. Однако этот инструмент поддерживает автодополнение, так что можете просто нажать на клавишу Tab, введя часть имени, и он предложит вам все возможные совпадения на основе ключевых слов.

Если вы введете в *Gmrun* web-адрес, он автоматически откроет браузер на этой странице, и он ведет журнал предыдущих команд, так что можно осуществлять по ним поиск. Вы даже можете настроить его на автоматический запуск правильного приложения на основе расширений файлов. Можете также запустить инструменты CLI, нажав на Ctrl+Enter после ввода команды.



Gnome-Pie ★★★★★

Gnome-Pie, несмотря на ограниченное количество функций, отнюдь не плох. В отличие от большинства других программ запуска приложений, он задуман для работы с мышью. Хотя вы можете его запустить с помощью настроенной комбинации клавиш, все же придется использовать мышью, чтобы решить, какие приложения вы хотите вызвать. Еще одно нарекание по поводу приложения в том, что оно не поддерживает поиск по рабочему столу, а значит, применимо только для запуска установленных приложений. Он предоставляет круговые интерфейсы для быстрой навигации, и каждый такой интерфейс называется Pie [англ. пирог]. По умолчанию он предоставляет Pie для Applications, Bookmarks, Main Menu, Multimedia, Session и Window, и вы можете вызвать любой из этих пирожков преднастроенными сочетаниями клавиш. Можно создать и собственный Pie, например, для наиболее часто используемых файлов, директорий или web-страниц.

Поддержка и документация

И кого же вам позвать, охотники до GNU?

Один из проверенных временем способов овладеть новым приложением — метод проб и ошибок; в ином случае можно прочитать документацию или обратиться за помощью к сообществу, если возникнут проблемы.

К сожалению, это область, в которой большинство приложений из нашего списка ничем особенным похвастаться не могут. За исключением *Kupfer*, который предоставляет качественную документацию с подробным освещением всех граней программы на нескольких языках,

остальные программы запуска рассчитывают, что вы все выясните сами.

У некоторых проектов, как, например, *Gmrun* и *Gnome-Do*, по сути нет даже официального сайта: *Gmrun* имеет страничку на Sourceforge, а *Gnome-Do* — на Launchpad. Страница проекта *Gnome-Do* даже не упоминает, что надо установить отдельный пакет, если требуются какие-то плагины. Впрочем, следует учесть, что поскольку проекты *Kupfer* и *Gnome-Do* похожи, большинство документации для *Kupfer* с тем же успехом применимо к *Gnome-Do*.

У *Gmrun* имеется map-страница, что несколько утешает, и глобальный файл настройки `/etc/gmrunrc` весьма информативен; а поскольку инструмент широко используется, легко найти множество обсуждений на разных форумах. То же касается и *Homerun*. Его официальная документация, мягко говоря, поверхностна, но есть уйма пользовательских блогов, где обсуждается установка и настройка в разных дистрибутивах. Сайт *Gnome-Pie* дает инструкции по использованию плюс видео, так что вы сможете увидеть его в действии.

Вердикт

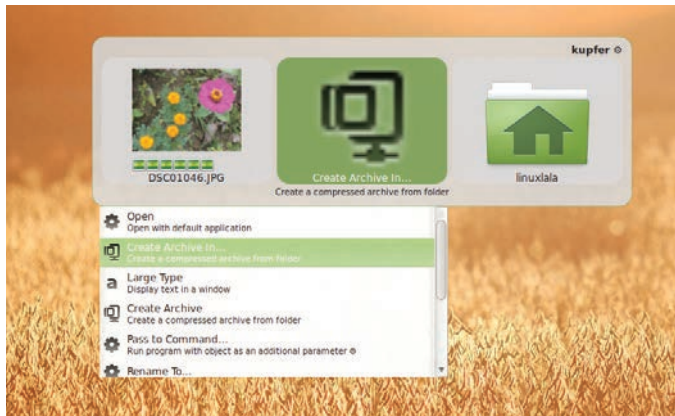
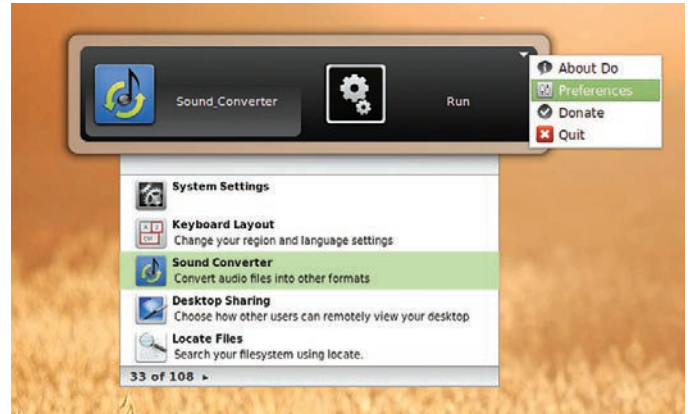
- Kupfer** ★★★★★
- Gnome-Pie** ★★★★★
- Homerun** ★★★★★
- Gmrun** ★★★★★
- Gnome-Do** ★★★★★

» Лучший способ найти ответ на проблемы — обратиться к форумам вашего дистрибутива.

Gnome-Do ★★★★★

Вы можете настроить *Gnome-Do* на запуск при входе в систему и затем получать доступ к нему через область уведомлений или с помощью сочетания клавиш Super (Windows Key) + пробел. Можете изменить это сочетание, чтобы запускать *Gnome-Do* в диалоговом окне Preferences.

На первый взгляд интерфейс может показаться странным, но работает гладко. Начните вводить текст, и он автоматически предложит все результаты совпадений, будь то приложения или файлы. Однако можно сначала включить плагин Files and Folders и предоставить список директорий для индексации, чтобы *Gnome-Do* заодно служил хорошим инструментом поиска по рабочему столу. Все результаты совпадений отображаются без особого промедления, и их можно выбрать с помощью стрелок. Выбрав файл или приложение, нажмите на клавишу Tab, чтобы увидеть все действия, связанные с этим выбором.



Kupfer ★★★★★

Gnome-Do и *Kupfer* выглядят похоже и предлагают похожий набор функций, и для обоих источником вдохновения послужил инструмент Mac OS X *Quicksilver*. Однако *Kupfer* больше подходит для новых пользователей, так как предлагает более удобные подсказки и документацию. При работе с изображениями и файлами PDF, *Kupfer* также сразу предлагает миниатюры файлов. Этой функции нет в *Gnome-Do*. *Kupfer* также запоминает последнюю использованную директорию и открывает ее при следующем запуске программы. Если вам нужен новый поиск, можете не обращать на это внимания и вводить текст, или продолжить просмотр этой директории.

Чтобы добавить к Favourites файл, приложение или директорию, сначала выберите его в списке, нажмите Tab и затем выберите действие Add to favourites. *Kupfer* выдает Favourites в качестве первого предположения, как только вы наберете fa.

Homerun ★★★☆☆

Есть два способа запуска *Homerun* на вашем рабочем столе KDE: использовать его взамен меню по умолчанию или как полноправный рабочий стол. Наверху размещаются четыре вкладки: Home, Applications, Files и Power, и первая выдает список ваших любимых приложений и недавно использованных документов. По щелчку на приложении оно запустится, однако вы можете щелкнуть по нему правой кнопкой, чтобы определить его как любимое приложение. При просмотре директорий *Homerun* будет открывать папки, не вызывая менеджер файлов. Можно искать файлы и приложения с помощью панели поиска наверху справа, но поиск ограничивается выбранной вкладкой. Например, если вы находитесь во вкладке Power, и ищете *Kmail*, никаких результатов вам не покажут. Из всех инструментов в нашем Сравнении этот был самым медленным и вялым. Периодически он переставал отвечать, и не поминал никаких нажатий на клавиши.



Клавиши быстрого запуска

Когда вам надоест мышь...

При правильно настроенных клавишах быстрого запуска идея заключается в том, что вам вообще незачем трогать мышь для работы с большинством программ запуска приложений.

Gnome-Pie, хоть и задуман для работы с мышью, позволяет настроить сочетания клавиш быстрого запуска для запуска разных Pie. Можно также запускать приложения с клавиатуры, но это не так интуитивно, как у других соперников. При запуске настроенного Pie нажатие на Alt+Ctrl покажет вам цифры для

каждого приложения в Pie. Можете затем нажать на Alt+Ctrl и цифру, чтобы запустить приложение.

Работая с *Gmrun*, вы нажимаете на клавишу !, чтобы перейти в режим поиска. Загляните в файл `/etc/gmrunrc`, чтобы узнать, какие сочетания клавиш уже настроены и как они работают. Затем можете создать собственные, создав файл `~/.gmrunrc` и добавив их туда. Введите двоеточие после сочетания клавиш, затем ключевое слово, и *Gmrun* будет использовать указанное приложение для работы с этим ключевым

словом. Например, `man:less` будет автоматически запускать терминал и открывать man-страницу для `less`.

Для *Gnome-Do* и *Kupfer* щелкните правой кнопкой по значку в области уведомлений, нажмите Preferences и выберите вкладку Keyboard. Вы увидите список всех преднастроенных сочетаний клавиш. Чтобы изменить любую из них, нажмите Enter и введите собственные сочетания. Когда это заработает, вы сможете вызывать все приложения с помощью сочетаний клавиш.

Вердикт

- Gnome-Do ★★★★★
- Kupfer ★★★★★
- Gmrun ★★★★★
- Gnome-Pie ★★★☆☆
- Homerun ★★☆☆☆

» Homerun не полнофункционален, и в нем много всего сочетаний клавиш.

Индивидуальная настройка

Изм-м-м-менения. Придется повертеться.

Не все инструменты в этом Сравнении верят в выбор для пользователя, когда речь идет о настройке внешнего вида программы запуска. Из пятерки в нашем списке только *Gnome-Do* и *Gnome-Pie* поставляются с разными темами.

Для *Gnome-Do* вы переходите в диалоговое окно Preferences и щелкаете там по вкладке Appearance. На выбор вам предоставят четыре преднастроенные темы.

При выборе тема в действии не показывается, и чтобы увидеть изменения, сначала придется закрыть диалоговое окно Preferences, а затем запустить приложение. Помимо тем, по умолчанию *Gnome-Do* предлагает вид Classic, где можно изменить цвет фона.

Подобным же образом, *Gnome-Pie* предлагает ряд тем для выбора, демонстрируя их по щелчку правой кнопкой на значке панели и затем на Preferences.

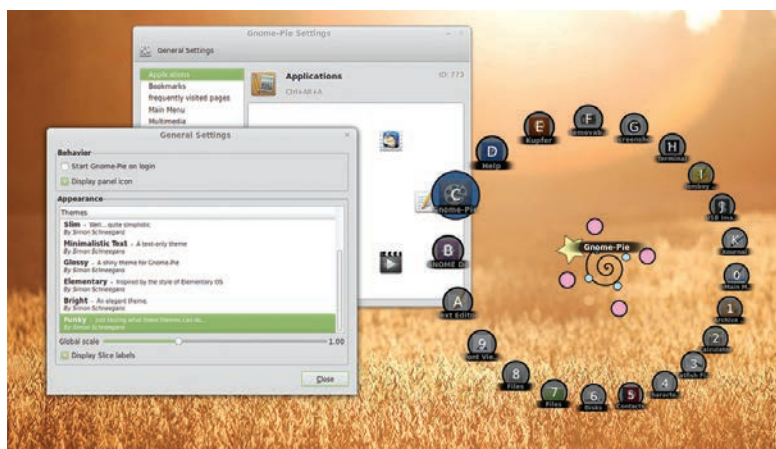
После чего нажимайте на кнопку General Settings, расположенную сверху слева.

Предлагая девять тем, *Gnome-Pie* надеется угодить всех пользователей. Каждый пункт в настроенном Pie называется Slice, и вы можете добавить любое количество Slice в существующие Pie или создать новые Pie. Для создания новых Pie, щелкните + внизу слева диалогового окна Settings.

Со своим минималистским интерфейсом, *Gmrun* не предоставляет графической возможности для изменений. Вы должны перейти в файл настройки /etc/gmrunrc или создать индивидуальный для пользователя файл ~/.gmrunrc. И даже тогда единственным визуальным изменением, которое вы сможете сделать, будет изменение ширины окна.

Gmrun, *Kupfer* и *Homerun* строят свой внешний вид на текущей теме и значках вашей среды рабочего стола. Для *Kupfer* и *Gmrun* вы должны изменить значки и тему GTK вашего рабочего стола.

Поскольку *Homerun* — приложение KDE Plasma, оно использует диалоговые окна и фон элементов Plasma для придания цвета фону. Чтобы изменить его внешний вид, можете изменить тему.



➤ Основная тема Gnome-Pie кажется перебором, но ее легко полюбить.

Вердикт

- Gnome-Pie ★★★★★
- Gnome-Do ★★★★★
- Gmrun ★★☆☆☆
- Homerun ★★☆☆☆
- Kupfer ★★☆☆☆

» Менять всю тему рабочего стола ради настройки одного приложения — явный перебор.

Наличие плагинов

Как получить от программ запуска еще больше.

Мы вскользь упомянули в этом сравнении несколько функций, но не все они доступны прямо сразу, и кроме *Gnome-Pie*, все остальные программы предлагают для расширения функциональности плагины.

У *Gmrun*, строго говоря, плагинов нет, но его структура придает ему невероятную гибкость. Его легко заставить работать с любым количеством приложений, отредактировав файл ~/.gmrunrc. Например, вы можете велеть *Gmrun* автоматически запускать VLC по определенному расширению файла, добавив такую строчку к файлу:

```
EXT:mpeg,mpg,avi,mkv,flv = vlc %s
```

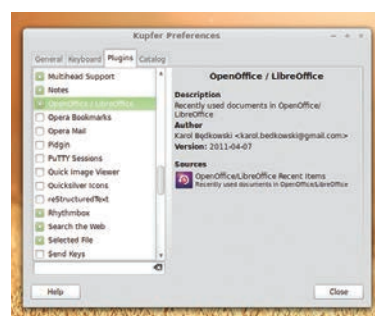
Аналогично можно настроить другие приложения на основе расширений файлов. Можете также установить *Catfish* и затем настроить *Gmrun*, чтобы использовать его для поиска по рабочему столу:

```
URL_srh = catfish --hidden 'path=~/%s'
```

и потом, при вводе, например, `srh: searchterm` *Gmrun* начнет искать term в указанном месте.

Homerun предлагает несколько плагинов под названием Sources, которые облегчают доступ к контенту. Щелкните по кнопке наверху справа и выберите Configure Tabs. Для каждой из четырех вкладок по умолчанию вы можете добавлять плагины и даже создавать собственные вкладки с собственными плагинами.

Как и практически во всем, здесь *Gnome-Do* и *Kupfer* тоже идут ноздря в ноздю. Оба они предлагают большой выбор плагинов. В любом из них перейдите в диалоговое окно Preferences и щелкните по вкладке Plugins. По щелчку на плагине из предлагаемого списка *Kupfer* предоставит подробную информацию о том, что он делает, и покажет все его настраиваемые параметры.



В *Gnome-Do* каждый плагин имеет внизу небольшой текст — описание того, что он делает. Например, плагин *Firefox* позволяет осуществлять поиск по закладкам и журналу. Однако не все плагины работают, как было обещано, а иногда даже способны спровоцировать неисправности в программе. Причем многие плагины в подобных случаях создавали сбои без всяких сообщений об ошибках.

➤ Учтите, некоторые плагины *Kupfer* и *Gnome-Do* обвешаны зависимостями.

Вердикт

- Gmrun ★★★★★
- Gnome-Do ★★★★★
- Kupfer ★★☆☆☆
- Gnome-Pie ★★☆☆☆
- Homerun ★★☆☆☆

» Как ни печально, плагины для *Homerun* не добавляют крутой функциональности.

Пускочки приложений Вердикт



Unity Dash, Gnome Shell и современные меню в большинстве дистрибутивов Linux ликвидировали необходимость в подобных специализированных программах запуска приложений. Единственная причина, по которой вам, возможно, понадобится их использовать — это дополнительные функции, которые они предлагают, не создавая нагрузки на системные ресурсы.

За исключением *Homerun*, мы не столкнулись с проблемами производительности в любой из этих программ запуска, и они преспокойно работали у нас сутками.

Из всех инструментов в списке, *Homerun* — единственный на стадии активной разработки. Однако ему предстоит долгий путь, прежде, чем он сможет конкурировать с Dash, или с Gnome Shell, или с другими инструментами в нашем Сравнении.

Для *Gnome-Pie* не нашлось места на пьедестале, но поясним: с ним все хорошо. Просто, по сравнению с тремя лучшими, он не предлагает таких же функций.

Если бы не *Gmrun*, мы бы присудили первое место *Kupfer*. Он невероятно легко устанавливается и предлагает отличную документацию, а отсутствие документации — единственное, что отличает *Gnome-Do* от *Kupfer*.

И хотя *Gmrun* уступил *Kupfer* и *Gnome-Do* практически по всем категориям, мы все же решили объявить победителем его. Это было действительно очень трудное решение. И обусловлено оно следующими причинами...

Внешний вид не кажется нам достаточным основанием не присуждать ему победу, поэтому нас не волнует отсутствие тем в *Gmrun*.

Плагины добавляют функций в *Gnome-Do* и *Kupfer*, но в то же время приводят к сбоям. А вот в *Gmrun* нет плагинов, но легко создать привязки клавиш, чтобы он начал работать с любым установленным приложением. Кроме того, его можно настроить, чтобы

он без особых хлопот осуществлял поиск по сети и рабочему столу.

Принимая во внимание большое количество пользователей, которые обсуждают *Gmrun* и делятся рецептами всяких хитрых трюков, отсутствие официальной документации не выглядит проблемой. Для нас вполне достаточно сообщества пользователей, готовых прийти на помощь друг другу.

Нам показалось, что по сравнению с другими инструментами, *Gmrun* требует несколько больших усилий в настройке, однако мы полагаем, что дело того стоит.

» Поскольку в *Gmrun* особо смотреть не на что, вот вам настройки *Homerun*.

«Пользователи обсуждают *Gmrun* и делятся рецептами хитрых трюков.»

I Gmrun ★★★★★

Сайт: SourceForge <http://bit.ly/M5MaJV> Лицензия: LGPL Версия: 0.9.2

» Несмотря на необходимость возиться с файлами настройки, явный победитель.

IV Gnome-Pie ★★★☆☆

Сайт: <http://bit.ly/1ghbGWQ> Лицензия: GPLv3+ Версия: 0.5.5

» Очень ограниченная программа запуска, но работает вполне гладко.

II Kupfer ★★★★★

Сайт: <https://github.com/engla/kupfer> Лицензия: GPL v3+ Версия: v208

» Проект уже несколько лет не видел новых релизов, но исправно работает.

V Homerun ★★★☆☆

Сайт: <http://userbase.kde.org/Homerun> Лицензия: GPL Версия: 1.1.0

» Стоит поработать над производительностью, а также над поиском и плагинами.

III Gnome-Do ★★★★★

Сайт: <https://launchpad.net/do> Лицензия: GPL v3+ Версия: 0.95.1

» Такой же удобный, как Kupfer, но без документации.

Обратная связь

Вы согласны с нашим выбором? Или сами работаете над проектом программы запуска? Шлите нам свои мнения на lx.f.letters@futurenet.co.uk.

Рассмотрите также...

Мы вполне сознательно не включили в наше Сравнение современные программы запуска типа Ubuntu Dash и Gnome Shell. И единственной причиной включить в него *Homerun* было желание представить здесь KDE.

Остальные инструменты в нашем списке можно установить в любой среде рабочего стола. Но если вам нужно легковесное приложение

для *Xfce*, попробуйте *Dmenu* или *Whisker Menu*. Из этих двух, *Dmenu* предлагает ограниченное количество функций, но существует дольше, и может похвастаться обширной базой преданных пользователей.

В данном Сравнении мы использовали ряд проектов, которые активно уже не разрабатываются. Мы прихватили *Kupfer* и *Gnome-Do* потому,

что у них по-прежнему остается немало поклонников. Точно так же многие до сих пор используют *Synapse*, который больше не разрабатывается. В нем практически такие же функции, что и в *Kupfer*.

Пользователи KDE несколько избалованы выбором. Помимо *Homerun*, здесь также есть *Krunner*, *Lancelot*, *Kickoff* и другие. Везунчики. **LXF**

ЗАБЬЕМ АНБ

Взбешенный
онлайн-слежкой,
Маянк Шарма
учит вас контроли-
ровать свою кон-
фиденциальность
онлайн.



За вами следят организации с трехбуквенными названиями и корпорации стоимостью в миллиарды долларов. Они читают ваши электронные письма и размещаемые вами комментарии, и видят публикуемые вами фото. Они знают, какие статьи вы читаете, какое видео смотрите и где предпочитаете совершать покупки. Эти люди прекрасно понимают, что слежка вам не нравится, но правда в том, что им на это плевать. За вашей онлайн-активностью следят во имя национальной безопасности и под прикрытием целевой рекламы.

Эта совершенно оруэлловская утрата конфиденциальности уходит корнями в необузданный обмен электронной информацией. Как такового, «Большого Брата» не существует. Зато вместо него существуют сотни «Маленьких Братьев», которые повсюду следуют за нами, когда мы используем и просматриваем Интернет.

Наша индивидуальная свобода угодила под удар технологии. Вы не можете закрыть глаза на эту слежку потому, что вам якобы «нечего скрывать»: разоблачения бывшего сотрудника АНБ Эдварда Сноудена [Edward Snowden] продемонстрировали тайные операции и всепроникающие базы данных, которые отслеживают всю вашу онлайн-активность независимо от того, преступник ли вы или законопослушный гражданин.

«Эти люди понимают, что слежка вам не нравится, но им на это плевать.»

Конфиденциальность не в том, чтобы что-то скрывать, а в том, чтобы контролировать, какую часть нашей жизни мы оставляем для себя и какой частью готовы поделиться с остальным миром; а законы в этом мире систематически

перепишутся с целью облегчить доступ к вашему персональным данным.

Мы славим каждую попытку, способную помочь сохранить нашу безопасность, и нам отвратительно, когда наши личные данные попадают в лапы корпораций, которые набивают сундуки, продавая их.

В данной статье мы рассмотрим ряд лучших инструментов для защиты нашей конфиденциальности онлайн. Мы покажем, какая информация утекает у вас между пальцев и как ею злоупотребляют. Вы также узнаете, как контролировать свою видимость в Сети и стать частным гражданином Сети. В том, что мы здесь покажем, нет ничего нечестного или незаконного. Статья приглашает осознать опасность утраты своей конфиденциальности и защитить себя от незаконной слежки, от тех, кто крадет личную информацию, и от правительственных организаций (неважно, деспотические они или нет).

Защита вашей информации и вашей конфиденциальности идут рука об руку, и все начинается с ограничения информации, предоставляемой web-компаниям. Далеко не всегда они ставят во главу угла ваши интересы, а некоторые из них печально известны тем, что продают или обмениваются вашей персональной информацией.

Проще всего отследить вас через IP-адрес. По нему сайт может определить географическое положение, то есть город или область. Это достаточно общая практика сетевой рекламы, которая пытается привлечь ваше внимание, упоминая ваше местонахождение. IP-адреса являются динамическими, и со временем становятся неподходящими для отслеживания пользователя. Но, сочетая ваш IP-адрес с другой информацией, например, HTTP-реферерами и куки, отследить вас легко.

Задача заголовка HTTP-реферера в том, чтобы загрузить сайт, по которому вы щелкнули, и проинформировать его, откуда вы. Он также отсылается при загрузке контента на web-странице. И если web-страница содержит рекламу, ваш браузер расскажет рекламодателю, какую страницу вы просматриваете. Некоторые нечистоплотные рекламодатели встраивают в сообщения электронной почты невидимые изображения, которые используют HTTP-реферер, когда вы открываете почту.

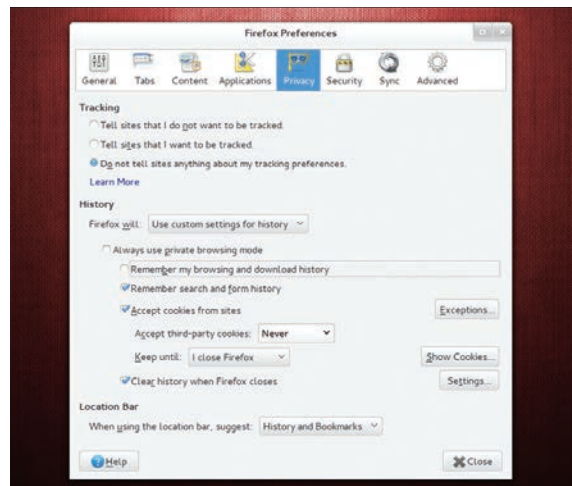
Крутые куки

Почти все сайты используют куки, чтобы хранить информацию о посетителях и о том, как они используют сайт. Куки хранятся на компьютере пользователя, однако у пользователя невелик контроль за тем, какая именно информация там содержится.

У куки есть множество вполне законных областей применения, например, хранение настроек и закладок на сайте: так, почтовые онлайн-сервисы употребляют куки для запоминания вашего логина.

Однако куки также позволяют сайту отслеживать ваши перемещения по нему. Пусть это и звучит довольно безобидно, однако крупные сайты, вроде Google, не сидят на одном домене. Google, как вы, вероятно, знаете, контролирует крупнейшую в Интернете рекламную сеть. Когда вы переходите с сайта на сайт, система рекламы, в дополнение к отображению рекламных объявлений, также отслеживает, какие сайты вы посещаете, и впоследствии использует эти данные для отображения рекламных объявлений, подобных тем сайтам, которые вы посетили.

Google не единственный, кто этим занимается: согласно исследованию www.digitaltrends.com, не менее 125 разных компаний или продуктов компаний отслеживают вашу онлайн-активность на 100 топ-сайтах. В основном это всего лишь рекламные сети, но есть и особо гнусные. Взять, например, виджет комментариев Disqus. Он отслеживает по сайтам, которые его используют,



такую информацию, как журнал постов пользователя, IP-адрес и версию браузера, даже если пользователь с сайта ушел. Также они размещают комментарии на страницах публичного профиля у каждого пользователя.

Некоторые публичные сервисы предусматривают опцию отказа от рекламы, но есть и такие, что навязывают рекламу насильно. Вы можете отписаться от основных отслеживающих сетей, зайдя на страницу Network Advertising Initiative's Opt-Out (www.networkadvertising.org/choices). Этот онлайн-сервис проверит вашу систему на предмет наличия отслеживающих куки от рекламных сетей и позволит вам от них отказаться.

Кроме того, все браузеры предлагают опцию удаления всех куки. Можете также воспользоваться режимом Private Browsing своего браузера, чтобы обеспечить стирание всех куки по закрытию окна. Режим также не позволяет сайту узнать, где вы были до этого, что особенно удобно, если вы используете общественный компьютер.

Однако есть одна подлая маленькая куки, гораздо более агрессивная, чем стандартные. Local Shared object (LSO) или Flash cookie, как ее обычно называют, исключительно опасна, потому что не хранится с другими куки и создана как раз во избежание обычно применяемого контроля конфиденциальности.

Чтобы ограничить то, как Flash хранит LSO, зайдите в менеджер онлайн-настроек Flash (<http://bit.ly/1m33E9X>) и уберите выделение с опции Allow Third-Party Flash Content To Store Data On Your Computer [Разрешить стороннему Flash-контенту хранить данные на вашем компьютере]. Учтите: пойдя путем ограничения использования куки, вы повлияете на работу своего браузера; однако сохранение конфиденциальности того стоит.

» Обязательно проверьте в Параметрах приватности Firefox, заблокированы ли сторонние куки.

Знаете ли Вы?..

АНБ собрало большое количество метаданных об интернет-трафике. Например, кто с кем разговаривает, когда, и долго ли. Метаданные намного проще хранить и анализировать, и они могут быть очень личными.

»

Переходите на SSL

Одно из первых действий, которое вам стоит предпринять, ступая на зыбкую почву Интернета, это перейти на протокол Secure Sockets Layer (SSL), с шифрованием вашего сетевого трафика. SSL использует сертификаты для создания безопасной, зашифрованной связи между браузером посетителя и web-сервером, который размещает страницу.

Шифрованное соединение сохранит на время передачи конфиденциальность любых ваших данных, передаваемых из браузера на web-сервер — например, информации о ваших кредитных картах.

Сертификат предоставляется соответствующим органом, например, VeriSign или Thawte. Все сайты с шифрованием по SSL отображают в браузере значок с замком, и щелкнув по этому значку, вы узнаете о сертификате более подробно.

Но следует знать о некоем подвохе. Сертификатов SSL есть несколько видов, и фишинговые сайты могут вполне законно таковой приобрести, чтобы убедить клиентов в своей надежности.

Проявляйте бдительность, приглядывая за сертификатами Domain Validated. Они недороги,

но не удостоверяют и не подтверждают подлинность компании, которая скрывается за сайтом. Нажав на значок с замком, вы не увидите никакой другой информации, кроме информации по шифрованию. Другие сертификаты безопасности предоставляют данные об организации, которой принадлежит сайт.

У каждого небезопасного сетевого протокола есть безопасный эквивалент: для просмотра сети — HTTPS, для передачи данных — SFTP и SCP, а для удаленного входа в систему — SSH.

Заметайте следы

Вот как можно посещать Сеть, не оставляя следов.

Знаете ли вы?..

По словам Эдварда Сноудена, мониторинг сетевой активности более эффективен, чем атаки на системы, поэтому у АНБ есть программы, которые проникают на пользовательские устройства, например, ноутбуки и роутеры, и превращают их в устройства слежения, которые можно включать удаленно.

Даже если вы предпринимаете все меры предосторожности, чтобы минимизировать свои следы в Интернете, и используете только сайты с шифрованием, вас все равно видно. Ведь вы предоставляете свой IP-адрес всем наблюдателям, включая посещаемые вами сайты.

Вдобавок не все сайты используют SSL, и вы все равно передаете важные сведения для входа в систему и прочую информацию по нешифрованным каналам. А ее легко перехватить с помощью таких инструментов анализа пакетов, как *Wireshark*, особенно через небезопасные сети, вроде общественной точки доступа Wi-Fi. Есть ряд решений, помогающих замести следы и замаскировать свои цифровые отпечатки, обойти цензуру и остаться невидимым онлайн. Это бывает удобно, потому что некоторые сайты блокируют доступ посетителям из нежелательных для них государств.

Наиболее популярный способ — это Virtual Private Network [Виртуальная частная сеть] или VPN. Ее главная цель — развернуть частную сеть поверх публичной, позволив удаленным сотрудникам создавать соединения и использовать сервисы в сети своего офиса. Те же функции делают ее идеальным инструментом для создания безопасного соединения с Интернетом и гарантии того, что все ваши данные, и отсылаемые, и получаемые, зашифрованы и защищены от любопытных глаз.

«Многие сервисы VPN ведут журнал и согласны сотрудничать с правоохранителями.»

Сервисов VPN имеются десятки; большой перечень приведен в *wiki* о цензуре в Интернет, <http://en.cship.org/wiki/VPN>. При выборе VPN убедитесь, что она работает не только на уровне приложений. Есть, например, VPN, работающие только внутри веб-браузера. Их недостаток в том, что они защищают только то, что в браузере. Если вы должны запустить вместе с *Firefox* другой браузер или отдельную почтовую программу, то данные из этих других программ не будут защищены.

Плагины конфиденциальности

» **BetterPrivacy** предлагает вам всегда удалять все локальные объекты общего пользования (LSO) при закрытии браузера.
 » **HTTPS Everywhere** Заставляет браузер использовать HTTPS со всеми сайтами, которые его поддерживают.
 » **The Web of Trust** Определяет опасные сайты в результатах поиска.
 » **DoNotTrackMe** Не позволяет третьим лицам, рекламным агентствам и поисковым машинам отслеживать, какие веб-страницы вы посещаете.

» **Disconnect** Запрещает отслеживание более 2000 известных трекеров.
 » **Priveazy Lockdown** При посещении сайта, поддерживаемого плагином, тот предлагает некоторые задачи, которые вы должны выполнить, чтобы обеспечить защиту своей конфиденциальности. Когда вы щелкаете по задаче, *Priveazy* автоматически загружает соответствующую страницу настройки вместе с подробными инструкциями по изменению определенных настроек.



» Интерфейс *JonDo* включает *AnonymOMeter*, измеряющий уровень анонимности, предлагаемый активным сервисом.

Некоторые VPN могут также ограничивать определенные сервисы, такие, как одноуровневые сервисы обмена файлами типа *BitTorrent*. Также многие сервисы VPN ведут журнал и заявляют, что согласны сотрудничать с правоохранительными органами при наличии нужных документов. Есть чудесное описание сервисов VPN, которые серьезно подходят к вопросу анонимности, от [TorrentFreak.com](http://bit.ly/1dvMqay) (<http://bit.ly/1dvMqay>).

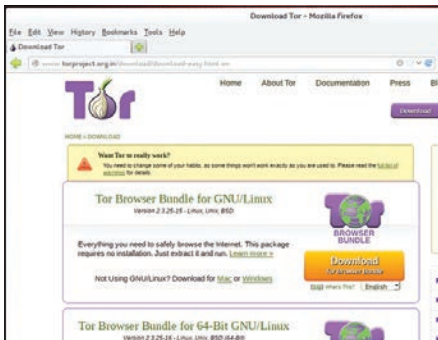
Если вы ищете VPN, ищите сервис, который поддерживает OpenVPN и использует SSL/TLS для обмена ключами. Пользователи, озабоченные проблемами конфиденциальности могут также захотеть выбрать сервис, который работает за пределами их страны. Сервис, который имеет серверы во многих местах, всегда является лучшим выбором.

Шелушите луковицу

Другой способ обойти цензуру и сохранить свою анонимность — использовать средства прокси-сервера. Самый популярный из них — сеть *Tor*. *Tor* (акроним от The Onion Router [Лукочичный маршрутизатор]) — это программа, создающая сеть, которая позволяет людям анонимно посещать Web. *Tor* создает сеть из транзитных узлов в Интернете. Когда вы заходите на сайт через *Tor*, данные, передающиеся на ваш компьютер и с него, собираются в этих узлах до того, как попасть на сайт, что маскирует ваше местоположение для сайта. Вы можете использовать *Tor* для посещения сайтов, которые засекают посетителей по их географическому положению. Простейший способ использовать *Tor* — через его браузерный комплект [Browser Bundle] для соединения с сетью *Tor* (См. *Устанавливаем Tor Browser Bundle*, стр. 37).

Недостаток *Tor* в том, что сайт грузится медленнее, поскольку сетевым данным приходится проходить через множество транзитных узлов. Далее, некоторые ISP, в частности, в Китае, активно ищут и блокируют узлы *Tor*, затрудняя соединение для некоторых пользователей. Помните также, что *Tor* шифрует только трафик с вашего компьютера до выходного узла, что не позволяет вашему ISP следить за вами. Но поскольку трафик на выходном узле не зашифрован, любой работающий на выходном узле может видеть ваш интернет-трафик. Есть неподтвержденные сообщения о том, что многие выходные узлы управляются правительственными агентствами.

Устанавливаем Tor Browser Bundle



1 Скачайте и запустите

В Bundle есть все, что нужно для соединения с сетью Tor, включая *Tor Browser*, индивидуальную версию *Firefox*. Распаковав его, перейдите в директорию в CLI, чтобы запустить скрипт `.starttorbrowser`.

2 Работайте анонимно

Этот скрипт запускает *Vidalia Control Panel*, которая соединится с сетью Tor. После соединения запустится *Tor Browser*, направленный на <http://check.torproject.org>, который подтвердит вашу анонимность.

3 Просмотрите сеть

Щелкните по View the Network в *Vidalia*, чтобы вывести карту мира, которая покажет маршрут вашего трафика и активные транзитные узлы. По окончании работы после закрытия окон кэш браузера будет автоматически очищен и соединение будет завершено.

Один из способов устранить уязвимость на выходном узле Tor — применять только безопасные протоколы (HTTPS, SSH и т.п.). Пригодится также Java Anonymous Proxy [Анонимный прокси Java] под названием *JonDo* — для сокрытия вашего IP-адреса он употребляет взаимосвязанные прокси-серверы. *JonDo* похож на Tor, но главное отличие состоит в том, что он выбирает в качестве узлов только сертифицированных партнеров. Можно также назначать, через какие узлы прокси направлять трафик. Вы можете просмотреть расположение его прокси и сделать свой выбор для увеличения безопасности. У бесплатных пользователей *JonDo* ограничивает скорость соединения, но предлагает подписку на премиум-сервис, обеспечивающий такую же скорость, как сервисы VPN. Проект также подробно информирует, как оплатить сервис, не раскрывая своей анонимности.

JonDo, я тебя знаю

[Традиционно, Джон Доу — обозначение «любого неизвестного», — прим. пер.]. Чтобы использовать сервис, скачайте Java-клиент *JonDo*, распакуйте его содержимое и запустите от имени root его скрипт установки. Скрипт установит клиент в `/usr/local`. Затем можно его запустить, введя команду `jondo`.

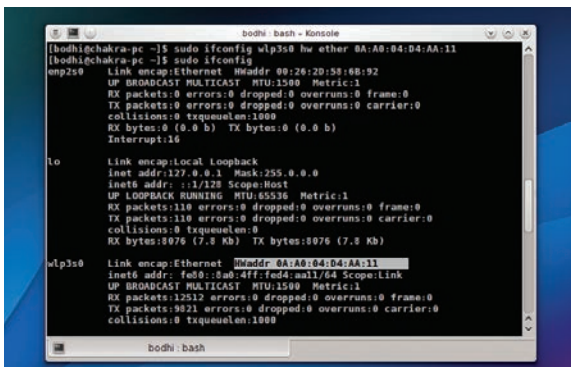
При первом запуске мастер установки проведет вас через короткий процесс соединения. По его завершении приложение

соединится с прокси-сервером. Прокси-сеть выбирается из выпадающего списка. Географическое положение каждой сети отмечено флажком ее страны.

В дополнение к инструменту *JonDo*, проект также производит безопасный профиль для *Firefox* под названием *JonDoFox*. Или скачайте собственный браузер *JonDo* на основе *Firefox*, под названием *JonDoBrowser*. Можете скачать и установить Deb-пакет для этого браузера с сайта проекта или добавить их репозиторий в свой дистрибутив на базе Debian. *JonDoBrowser* настроен для работы с прокси *JonDo*. Более того, в отличие от *Tor*, вы можете использовать приложение *JonDo*, чтобы отключить анонимность и продолжать использовать *JonDoBrowser*.

JonDo и *Tor* поладят и с другим браузером или сетевым приложением, например, приложением для обмена быстрыми сообщениями или клиентом электронной почты. Надо всего лишь сделать так, чтобы те направляли через *JonDo* и *Tor* весь свой трафик. Для этого перейдите на страницу настройки соединения приложения и укажите ручные настройки прокси. У *JonDo* это прокси SOCKSv5, хост 127.0.0.1 и порт 4001. Чтобы направить трафик через сеть Tor, используйте порт 9150, если работаете с Bundle.

Помните также, что бесплатные пользователи *JonDo* могут соединяться только через порты, применяемые для просмотра Сети, 80 для HTTP и 443 для HTTPS. Иначе придется подписаться на сервисы Premium. Хотя *JonDo* и *Tor* трудно сравнивать, многие считают первый более безопасной опцией. *Tor* более подвержен внутренним атакам, когда оператор узла сам атакует сеть. Возможность подобных атак в *JonDo* меньше, поскольку он защищает свои прокси.



» Обеспечьте, чтобы вас не отследили, замаскировав MAC-адреса вашей сетевой карты, например, `ifconfig eth0 hw ether 0A:A0:04:D4:AA:11`.

Дополнения системы безопасности

Помимо использования прокси-сервера, неплохо также оснастить ваш браузер плагинами безопасности и расширения конфиденциальности.

С помощью Adblock Plus вы можете помещать в черный и белый списки определенных рекламодателей. Еще одно

полезное дополнение — NoScript Security Suite, которое не позволит JavaScript, Java, Flash, *Silverlight* и прочему исполняемому контенту запускаться. Это дополнение может предотвратить межсайтовый скриптинг, межзональную перепривязку DNS и взлом роутера.

Стоя под радаром

Руководство по безопасности почты, обмена сообщениями и чата.

Знаете ли вы?..

По файлам Snowdena, проект GCHQ EdgeHill (названный в честь первой битвы Гражданской войны в Англии) надеется к 2015 году дешифровать программы, используемые 15 основными неназванными интернет-компаниями и 300 VPN.

Общаясь с другим человеком, вы выдаете гораздо больше информации о себе, о человеке, с которым разговариваете, и о природе вашей беседы. Единственный способ перехитрить слежку — шифровать свои разговоры. Тогда, даже если они будут перехвачены, их содержание узнать не смогут.

PGP (или Pretty Good Privacy — Отличная конфиденциальность) — самый популярный механизм шифрования данных и разговоров. Один из методов, которые PGP использует для шифрования, называется Public Key Cryptography. Не полагаясь только на пароль, этот метод использует сочетание публичного и приватного ключей для шифрования и дешифровки данных.

Суть такова: отправитель использует публичный ключ получателя для шифрования сообщения. Публичный ключ, как и подразумевает его название, должен быть доступен для всех. Получив сообщение, адресат использует свой хранимый в безопасном месте приватный ключ для дешифровки сообщения. Дополнительно отправитель также может зашифровать сообщение своим приватным ключом. Это помогает удостовериться личность отправителя сообщения. Получатель может удостовериться подпись с помощью публичного ключа отправителя.

может взять публичный ключ контактных лиц из преднастроенного списка серверов ключей и также создать индивидуальные правила для контактных лиц, чьи публичные ключи у вас есть.

Шифруйте свою почту

По умолчанию *Enigmail* создаст 2048-битный ключ, действительный пять лет. В дополнение к *Enigmail* можно также установить расширение для отзыва сертификата, чтобы признать частный ключ недействительным, если тот будет скомпрометирован.

Создав свои ключи, вы должны экспортировать их для безопасного хранения. Перейдите в меню OpenPGP и выберите Key Management [Управление ключами]. Щелкните правой кнопкой по ключу, который вы хотите хранить, и выберите опцию Export Keys to File [Экспорт ключей в файл]. Во всплывающем окне выберите опцию Export Secret Keys [Экспорт секретных ключей], чтобы сохранить публичный и приватный ключи в файле. Вы никогда и никому не должны пересылать этот файл. Имеет смысл хранить его в зашифрованном хранилище (см. *Создаем и используем том TrueCrypt*, стр. 41). Этот файл поможет вам импортировать ключи в другую систему, или восстановить их в случае потери.

Чтобы зашифровать сообщение ключами, создайте новое сообщение и выведите меню OpenPGP из окна Compose. В нем будут опции Sign Message [Подписать сообщение] и Encrypt Message [Зашифровать сообщение]. Будет целесообразно выбрать обе и затем продолжить писать свое сообщение, как обычно. При щелчке по кнопке Send [Отправить] у вас спросят парольную фразу. Введите ее, и ваше сообщение превратится в нечитаемый зашифрованный текст. Кроме самого сообщения, *Enigmail* умеет также шифровать любые вложения. Просто вложите файлы, как обычно, а *Enigmail* позаботится об остальном.

Ну, а для сервиса webmail, например, Gmail или Yahoo Mail, при отправке зашифрованных сообщений можно использовать плагин *Mailvelope* для браузеров *Firefox* и *Chrome* (см. *Шифруем webmail*, стр. 39). Плагин использует *OpenPGP.js*, библиотеку JavaScript, реализующую возможности OpenPGP в web-браузерах.

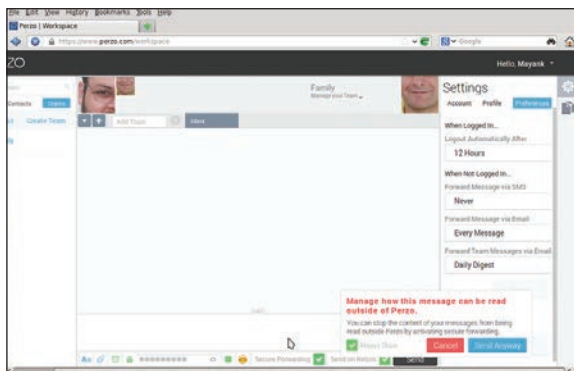
Одно лишь шифрование электронной почты не защитит ее от решительно настроенных агрессоров. Правоохранительные органы и другие заинтересованные лица могут заставить вас предоставить им ключи. В порядке дальнейшей предосторожности вы можете направить зашифрованную почту через прокси-сети *Tor* или *JonDo*. Расширение *TorBirdy* настроит *Thunderbird* на создание соединений в этих прокси-сетях.

Подобным же образом можно зашифровать сообщения в реальном времени, например, обмен быстрыми сообщениями. Два самых популярных клиента IM с открытым кодом, *Kopete* и *Pidgin*, могут шифровать сообщения через плагины. Например, протокол Off-The-Record (OTR) позволяет сквозное шифрование для общения IM. Оно реализовано через плагин *OTR*, который вы найдете в репозиториях большинства дистрибутивов. Для обмена зашифрованными сообщениями этот плагин должны установить обе стороны.

Когда вы установите и активируете плагин, вам придется сгенерировать приватный ключ для каждой настроенной учетной записи. И после этого, когда бы вы ни соединились с другим контактным лицом с данной учетной записи, плагин будет

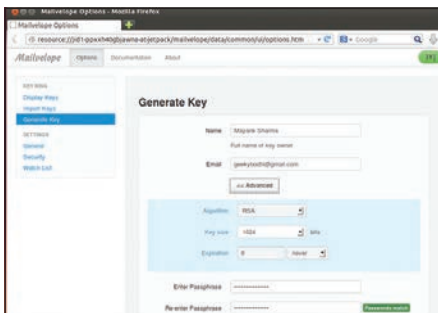
«Чтобы не допустить атаки посредника, ZRTP использует механизм SAS.»

Свободно доступный GNU Privacy Guard, широко известный как GPG — это альтернатива пакету программ шифрования PGP под лицензией GPL. Он предустановлен практически в любом дистрибутиве Linux. Многие настольные дистрибутивы Linux идут с почтовым клиентом *Thunderbird*. Его расширение-плагин *Enigmail* предоставляет преимущества GPG клиенту электронной почты, и его можно скачать в самом приложении. *Enigmail* автоматически запустит мастер настройки для конфигурации вашего расширения, который изменит параметры вашей электронной почты так, чтобы вы, скажем, не могли создавать сообщений HTML. Мастер можно и пропустить, и настроить *Enigmail* вручную. После установки расширение добавит новый пункт OpenPGP в меню *Thunderbird*. Не у всех ваших контактов есть публичный ключ, так что включить шифрование по умолчанию нельзя. *Enigmail*



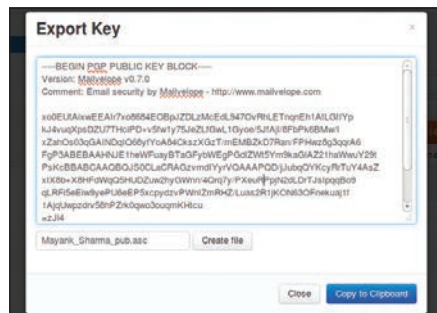
➤ **Perzo.com** — новый онлайн-сервис конфиденциальности. Кроме шифрования, он позволяет отправлять сообщения, уничтожаемые после заданного времени.

Шифруем webmail с помощью Mailvelope



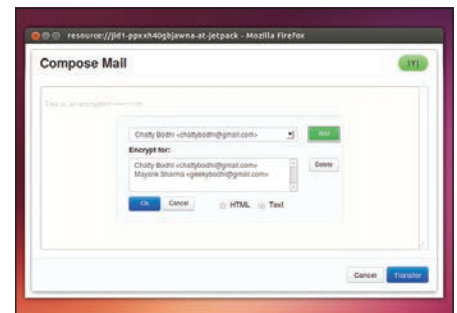
1 Установим Mailvelope

Инструмент *Mailvelope* для браузера *Chrome* можно установить из Play store, а дополнение к *Firefox* в данный момент на стадии разработки и доступно только из GitHub. Установив *Mailvelope*, найдите инструмент на панели дополнений *Firefox* и выберите в Options. Подробно Generate Key в панели навигации слева. Введите подробности, чтобы создать ключ для своего электронного адреса.



2 Обменяемся ключами

Прежде чем обмениваться зашифрованными сообщениями по электронной почте, ваши получатели должны добыть ваш публичный ключ. Чтобы отправить им свой публичный ключ, перейдите в Display Keys > Export > Display public key. Скопируйте и вставьте блок ключа и пошлите его электронной почтой всем своим друзьям. Вам же понадобятся их публичные ключи. Получив ключ, перейдите в Import Keys и вставьте ключ в окно, затем нажмите на Submit, чтобы добавить его к вашей связке ключей.



3 Зашифруем сообщения

Теперь создайте сообщение и нажмите на плавающий значок в окне. Введите сообщение и щелкните по значку с замком. Добавьте получателей, нажмите OK, и текст будет зашифрован. Затем нажмите на кнопку Transfer, чтобы вставить текст в тело сообщения. Когда сообщение будет получено, оно будет выбрано дополнением. Вашим получателям придется ввести пароль, чтобы прочитать сообщение.

автоматически создавать частное общение, если ваше контактное лицо должным образом настроило плагин *OTR*.

При использовании IM стоит помнить вот о чем: пусть разговор и зашифрован, но нет автоматического способа выяснить, разговариваете ли вы с вашим контактным лицом или с самозванцем. Плагин может помочь вам установить личность вашего собеседника посредством одного из его трех механизмов. Вы можете обменяться заранее обговоренным секретом или кодовой фразой, или задать вопрос, ответ на который известен только вашему собеседнику, или вручную удостоверить «отпечатки пальцев» вашего ключа с помощью другого способа коммуникации.

Плагин *OTR* создан для шифрования сессий чата один на один. Если надо зашифровать многопользовательские чаты, перейдите на <http://crypto.cat>. *CryptoCat* — это онлайн-сервис, позволяющий обезопасить разговор. Он шифрует сообщения внутри самого браузера с помощью AES-256 и 4096-битных асимметричных ключей.

Защищенные от слежки чаты

Чтобы использовать сервис, сначала вы должны установить его в качестве дополнения или в Firefox, или в Chrome. Начиная разговор, сперва выберите имя раздела чата, который хотите создать, а также имя экрана. Получив информацию, *CryptoCat* создаст для вас ключи шифрования и раздел чата, и осуществит вход в него. Желаящие к вам присоединиться должны для этого тоже установить дополнение к браузеру, и затем просто ввести имя вашего раздела чата. Поскольку здесь нет защиты паролем и вы все используете псевдонимы, а не свои настоящие имена, *CryptoCat* предлагает механизм Q&A [вопрос—ответ] для удостоверения личности пользователей. Более того, через интерфейс вы можете менять свой статус и переключать уведомления рабочего стола и аудиоуведомления.

CryptoCat создан для облегчения многопользовательских групповых разговоров. Однако с членом группы можно беседовать и приватно. Помните также, что хотя ваши сообщения шифруются, соединение не анонимно, и вашу личность можно



Используйте встроенные инструменты клиентов чата, чтобы установить личность человека на другом конце до того, как передадите ему конфиденциальную информацию.

отследить. Чтобы не допустить этого, *CryptoCat* рекомендует использовать сервис через прокси-сеть *Tor*.

Как и с текстом, вы можете делать безопасные голосовые и видеозвонки через Voice over IP (VoIP). Чтобы обеспечить конфиденциальность сторон, создатель PGP Фил Циммерман [Phil Zimmerman] создал протокол ZRTP.

Этот протокол отвечает за согласование ключей между соединенными узлами и устанавливает между ними соединение SRTP, которое и занимается шифрованием. Большую часть функций реализует библиотека *GNU ZRTP*.

Чтобы не допустить атаки посредника, ZRTP использует механизм под названием Short Authentication String [Короткая строка аутентификации], или SAS. ZRTP определяет длину SAS четырьмя символами. Когда совершается звонок с использованием ZRTP, одна сторона читает первые два символа SAS, а другая — последние два. Оба значения должны совпадать. Неплохо будет сравнивать значения в начале звонка и потом периодически, с разумными интервалами.

Java-реализация библиотеки GNU ZRTP — VoIP-клиент с открытым кодом *Jitsi* [см. Сравнение LXF181, стр. 30]. Он поддерживает протоколы SIP и XMPP и может транслировать рабочий стол и устанавливать голосовые конференц-звонки.

Укрепите свой форт

Зашифруйте свое оборудование и заприте рабочую станцию.

Знаете ли вы?..

АНБ выделяет солидные ресурсы для проникновения на компьютеры — этим занимается группа Специального доступа [Tailored Access Operations, TAO]. Считается, что у TAO целый набор эксплойтов, и она может проникнуть на любой компьютер, независимо от того, используете ли вы Windows, Mac OS или Linux.

После всех наших рекомендаций тревожит мысль о том, что если АНБ захочет проникнуть на ваш компьютер, оно это сделает. Ну, то есть, если вы верите всем подробностям документов, разглашенных бывшим сотрудником и разоблачителем АНБ Эдвардом Сноуденом. Однако сделать ваш компьютер невосприимчивым к атакам несколько сложнее, чем попустительство ко взлому, и сильным паролем здесь не всегда отобьешься. Взломщик, возможно, просто сумеет перезагрузить его, ответив на ваш беспечно выбранный секретный вопрос.

Создание сложных паролей — вопрос фундаментальной важности для обеспечения конфиденциальности. Но управление паролями — вопрос не менее важный. Записывать пароли на бумаге крайне опасно, и далеко не все способны, подобно Шерлоку, создать себе чертоги разума для хранения сложных паролей. Поэтому-то вам и нужен инструмент управления паролями, подобный *KeePassX*.

KeePassX — это программа создания кодовых фраз и управления ими. Инструмент может за считанные секунды создавать исключительно сложные и очень специфические кодовые фразы. Кроме того, *KeePassX* пригоден и для управления зашифрованными кодовыми фразами, доступ к которым позднее можно

Рекомендуется использовать сочетания заглавных и маленьких букв, цифр и спецсимволов.

Мы рассмотрели инструменты, позволяющие шифровать разговоры и приложения до того, как они покинут ваш компьютер. Однако это вам не поможет, если ваш компьютер украден или взломан. Единственный способ обеспечить сохранность своих файлов — зашифровать носитель, где они размещаются. Многие дистрибутивы Linux, в том числе Fedora и Ubuntu, дают возможность зашифровать диск перед установкой дистрибутива. Такое называется Full Disk Encryption, и согласно Electronic Frontier Foundation (EFF), это наилучший способ обеспечить конфиденциальность данных даже для начинающих пользователей.

Бывают случаи, когда нужно перенести важные файлы на портативном устройстве, например, на диске USB. В подобной ситуации наличие зашифрованного жесткого диска вам не поможет: нужен инструмент создания портативных зашифрованных томов.

Шифрованные диски

Есть несколько опций для шифрования портативных дисков — например, *Disk Utility* в Gnome. Однако самая популярная опция, несмотря на недавние разногласия, это инструмент *TrueCrypt*.

TrueCrypt — кросс-платформенный инструмент с открытым кодом, который буквально на лету создает зашифрованные тома. Это означает, что данные автоматически шифруются перед сохранением и дешифруются после загрузки. *TrueCrypt* неоднозначен, потому что его 70000 строк кода никогда не проверялись. Чтобы исправить это, была проведена кампания по сбору средств, и на проверку кода пожертвовали более \$46000. Можете отслеживать проект на www.istruecryptauditedyet.com.

Вместо создания зашифрованной папки, *TrueCrypt* создает виртуальный диск внутри файла и шифрует его. Когда бы вы ни использовали *TrueCrypt* для дешифровки диска, он будет присутствовать в вашем менеджере файлов в виде жесткого диска.

«Единственный способ обеспечить сохранность файлов — зашифровать их носитель.»

получить с помощью кодовой фразы-мастера. Наличие мастера устраняет необходимость в запоминании множества кодовых фраз. Поскольку мастер дает доступ ко всем кодовым фразам, он должен иметь иммунитет к атакам по словарю, анализаторам паролей и прочим угрозам, таким, как клавиатурные шпионы.

Облачные сервисы, предлагающие конфиденциальность

Если вам нравятся сервисы, предлагаемые популярными облачными сайтами, но вызывает подозрение их политика конфиденциальности, вот несколько опций, которые предлагают те же услуги, но при этом уважают вашу конфиденциальность.

Первым идет наш фаворит, *OwnCloud*. Это инструмент распределенного доступа к файлам наподобие Dropbox, но с открытым кодом, и вы можете установить его на собственном оборудовании. Вы сами отвечаете за свои данные, доступ к которым вы получите отовсюду, как в Dropbox. Если вы предпочитаете командную строку, есть также *SparkleShare*, который передает данные по каналам SSH.

Можно также синхронизировать папки с помощью помощника *Gitannex*, с поддержкой GPG. Если у вас есть Raspberry Pi, приглядывайте за находящимся в разработке проектом arkOS, который позволит вам разместить собственный сайт,

электронную почту, социальную сеть, облачные учетные записи и прочие сервисы.

Если вам не хватает времени и знаний для создания собственного сервиса хранения, воспользуйтесь сторонними сервисами, которые предпринимают меры по обеспечению вашей конфиденциальности. Если вы хотите поделиться с другим частным документом, используйте сервис <http://securesha.re>. Он зашифрует документ на вашем компьютере до его загрузки на web-сервер. Затем он создает защищенный паролем самоуничтожающийся URL, который вы можете передать получателю. На его стороне файл скачивается и дешифруется локально.

Подобным же образом, если вы хотите поделиться изображениями, аудио и видео, можете воспользоваться сервисом <http://mediacru.sh>, который предлагает бесплатное хранение и тоже

уважает вашу конфиденциальность. Он использует протокол HTTPS, не хранит никакой информации о пользователях, не показывает рекламы и учитывает настройки вашего браузера DoNotTrack.



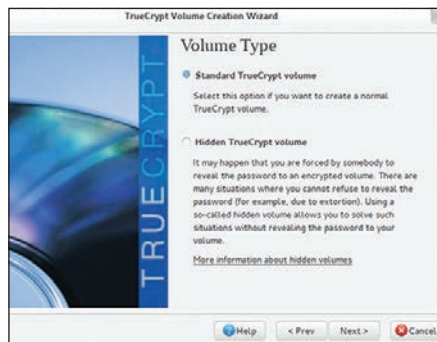
➤ Сервис *img.bi* — это сайт хостинга изображений, который шифрует файлы перед тем, как загружать их, и дает вам ссылку для последующего удаления.

Создаем и используем том TrueCrypt



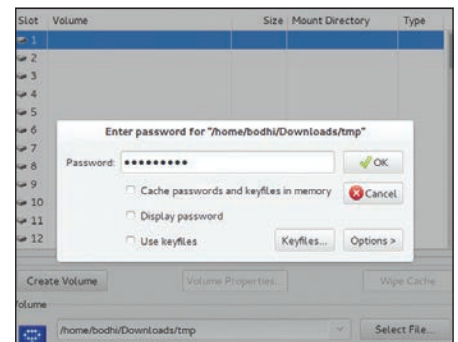
1 Создайте том

Запустите приложение и нажмите на кнопку Create Volume. Запустится мастер Volume Creation Wizard, который проведет вас по всем стадиям. Первый шаг позволяет вам выбрать, где будет создан том TrueCrypt. Начинающим пользователям рекомендуется первая опция.



2 Опции тома

Подобным же образом, для большинства последующих шагов выберите опции по умолчанию. У вас спросят, хотите ли вы создать обычный или скрытый том. Также вы должны указать имя файла, который будет выступать в роли зашифрованного контейнера, и его размер, а также алгоритм шифрования для тома.



3 Смонтируйте том

Снова на главном экране нажмите кнопку Select File и выберите зашифрованный том, который вы создали. Затем нажмите на кнопку Mount и введите пароль для тома. Том будет смонтирован и окажется в списке вашего менеджера файлов. Чтобы размонтировать его, нажмите на кнопку Dismount.

Когда он зашифрован, он появляется в вашей файловой системе как обычный файл. Вы даже можете скопировать его на другой компьютер и прочитать его содержание с помощью другого экземпляра TrueCrypt.

Если вам нужно действительно портативное решение шифрования, можете создать зашифрованный виртуальный диск внутри портативного диска и также запаковать в TrueCrypt скрипт оболочки Linux или исполняемый файл Windows. Это позволит вам дешифровать том TrueCrypt, где бы вы ни находились.

Один из главных недостатков шифрованных файловых систем — они не скрывают того факта, что вы что-то скрываете. Решительно настроенный взломщик может заставить вас передать ему ключи шифрования, или вас могут заставить это сделать по закону. В TrueCrypt есть возможность такое обойти: используя функции Hidden Volumes [Скрытые тома], вы можете создать зашифрованный том внутри другого тома. Вы взаимодействуете с внешним томом, как со стандартным томом TrueCrypt. Однако внутренний том скрыт, а место, которое он занимает, отображается как свободное место на диске.

Оба тома имеют разные пароли, и даже если вас принудили сообщить пароль для внешнего контейнера, внутренний том останется зашифрованным. А поскольку никто не знает о существовании внутреннего тома, вы можете прибегнуть к тому, что ЦРУ именуется правдоподобным отрицанием — а мы называем ложью.

Переключите дистрибутив

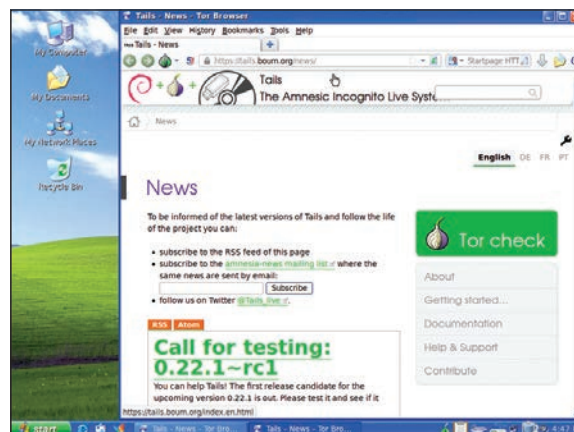
Если вы озабочены своей конфиденциальностью, следует отказаться от обычного дистрибутива и использовать один из специализированных дистрибутивов, разработанных для ее защиты. В них не просто нет функций, вторгающихся в вашу конфиденциальность (как, например, линзы Ubuntu Amazon): они содержат некоторые из инструментов, о которых мы рассказали в этой статье (см. Сравнение LXF174, стр. 24, где мы тестировали IpreidiaOS, Liberté, Privatix, Whonix и Tails).

Дистрибутив Tails (The Amnesic Incognito Live) на базе Debian работает из ОЗУ и тщательно следит за тем, чтобы не использовался раздел подкачки даже при его наличии. Выключаясь, Tails вычищает ОЗУ, чтобы там не осталось никаких следов использования. Tails использует рабочий стол Gnome 2, но предлагает

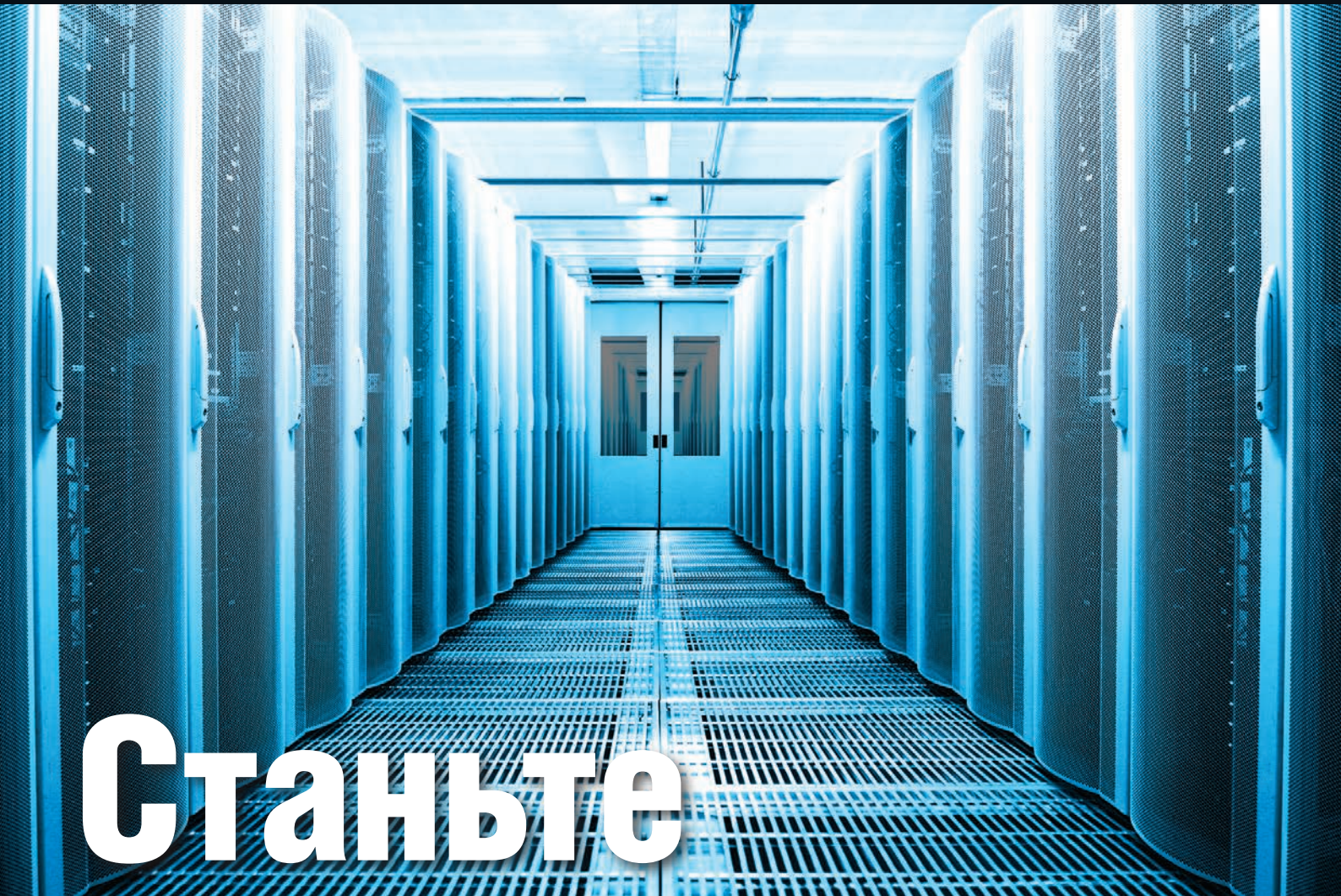
также опцию маскировки рабочего стола, чтобы тот выглядел как Windows XP. Дистрибутив включает Tor Browser Bundle, а его Tor Browser на базе Iceweasel содержит плагины расширения конфиденциальности, такие, как Https Everywhere, NoScript, Cookie Monster, AdBlock Plus и прочие.

Также вы найдете здесь менеджер паролей KeePassX, клиент Pidgin с плагином OTR и клиент электронной почты Claws, готовый отправить сообщения с GPG-шифрованием. Для создания шифрованных томов дистрибутив использует Disk Utility из Gnome.

Tails также включает виртуальную клавиатуру, чтобы защитить вас от клавиатурных шпионов. Менеджер файлов Nautilus оснащен расширением для безопасного стирания файлов и обеспечивает невозможность восстановления удаленных файлов с имеющегося пространства диска. Как мы уже указывали ранее, довольно трудно сохранять анонимность, особенно онлайн. Но если применять описанные в нашей статье инструменты, по отдельности, а лучше вместе, вы сможете неплохо скрыть свою личность и защитить себя от пронырливых правительственных агентств и коммерческих корпораций, выискивающих ваши данные. LXF



➤ Нет, мы не взяли скриншот Windows XP по ошибке. Это Tails в камуфляже, и им можно спокойно пользоваться в киберкафе, не вызывая излишнего любопытства.



Станьте профи Linux

Знатоки Linux в UKFast рассказывают о лучших возможностях карьеры в Linux и объясняют, как квалификация LPI влияет на жизнь сисадмина.

Даже для самого решительно настроенного человека попытка разобраться в инженерии Linux может оказаться весьма сложной. Принимая во внимание также отчет этого года от Linux Jobs, где говорится, что 9 из 10 работодателей считают найм профессионалов в сфере работы с Linux делом или «довольно трудным», или «очень трудным» (это выше, чем статистика прошлых лет), становится понятно, что необходим новый подход.

Linux играет важную роль в предоставлении услуг и техподдержке в UKFast, поэтому при компиляции, настройке и поддержке серверов в среде

времени, все же у нее есть ряд положительных моментов, которые превращаются в колоссальные преимущества на рабочем месте.

«9 из 10 работодателей считают найм профессионалов в Linux трудным делом.»

Linux крайне важно наличие хорошо подготовленных инженеров. И хотя сертификация LPI не превратит вас в сисадмина за сутки (на это нужно

Одно из ключевых преимуществ LPIC — стандартизация. Обучение дает вам базу, позволяющую работать с другими системами в уверенности, что вы сумеете найти способ самим справиться с разными проблемами.

Квалификация LPI постоянно прогрессирует и является отличным способом начать получать и развивать свои знания. LPIC1 предоставляет



» С квалификацией LPIC вы тоже сможете работать в таком месте, которое могло бы быть идеальным для воссоздания приключений Доктора Кто.



основные навыки и знания, общие для всех дистрибутивов Linux: системная архитектура, задачи администрирования и команды GNU и UNIX. LPIC2 основывается на них, и предлагает расширенные навыки для Linux-профессионала. Чтобы пройти этот уровень, инженеры должны уметь администрировать сайт от среднего до крупного размера. LPIC3 нацелен на инженеров уровня предприятия и предоставляет знания в специальных областях, таких, как смешанные среды и безопасность.

Хотя традиционный подход к обучению «за делом», когда старшие инженеры обучают «новичков», отлично работал, в UKFast обнаружили, что у разных наставников получаются разные результаты, в зависимости от успеха партнерских взаимоотношений. LPIC предоставляет куда лучший подход к изучению основ без обмена вредными привычками, и дает всем равные шансы овладеть знаниями и стать квалифицированным профессионалом Linux. Это позволяет компаниям обеспечивать стабильность для своих клиентов во всем.

Как и следует ожидать, являясь лидирующей компанией в области хостинга, UKFast получает множество резюме от инженеров, освоивших основы LPIC с помощью книг или обучающих видео онлайн. Чтобы получить квалификацию LPIC «любими

способами», нужен решительный настрой, и это доказывает, что вы имеете дело с человеком с высокой мотивацией. Конечно, UKFast не считает, что квалификацию обязательно нужно получать на дорогостоящем курсе. Они говорят, что чувствуют себя обязанными поделиться своим видением проблемы недостатка инженеров Linux, и сознают, что частично решение зависит от нас. Вместе с организаторами образовательных услуг они создали программу, чтобы получить инженеров, новичков и преданных сторонников, говорящих на одном языке. К счастью, их руководитель, Лоуренс Джоунс [Lawrence Jones], практикует передовой подход к обучению, и поэтому наш специализированный тренажер позволяет проходить обучение на месте.

Пока, естественно, это только начало, поскольку модель совершенно новая, но ответ на изначальный вопрос: «Можем ли мы взять 18-летних претендентов, не проходивших специального обучения, и использовать наш курс, чтобы провести их по LPIC?» явно будет уверенным «Да!» Если и другие компании присоединятся к UKFast в организации LPIC, а школы и далее будут обучать программированию и прочим компьютерным умениям, то, надо надеяться, станет возможно заинтересовать талантливых молодых людей в обучении и карьере, связанной с Linux. **LXF**

Один день из жизни юного сисадмина

Рбота инженером Linux может быть нелегкой, учитывая разнообразие проблем, с которыми вам приходится сталкиваться, и колоссальное количество команд и инструментов в вашем распоряжении. Мои функции заключаются в приеме звонков и заявок от клиентов по поводу их проблем с серверами, размещенными у нас; включая (но определенно не ограничиваясь) проблемы производительности, спама, вопросы по имеющимся обновлениям и жутких взломах.

Когда я только начал, я понял, что запуск определенной команды с определенными флагами может

решить определенную проблему, и испытывал искушение просто заучить их и использовать при необходимости. Это приводит к затруднениям, когда клиенты спрашивают, что вы на самом деле сделали и как они могут сделать это сами, или в случае, если команда не срабатывает, как ожидалось.

Возможность пройти сертификацию LPIC дала мне время вернуться назад и по-настоящему разобраться с целями разных флагов и альтернативами разным командам, позволив мне оценить, почему мы используем определенные команды, а не просто делаем то, что мне говорили.

Во время учебы в университете я проходил введение в Linux, но в основном курс был ориентирован на Windows. Однако это короткое введение позволило мне понять, что меня интересует карьера инженера Linux, и моя компания позволила мне получить мои знания на работе, в то же время повышая квалификацию, как с LPIC. Мне кажется, самой большой сложностью для тех, кто делает карьеру в Linux, может стать неосведомленность о Linux: когда я был в школе и принимал решения о своем будущем, я вообще не знал о Linux или о возможностях, которые Linux предоставляет.

Линуксоид

Билл Куинн, глава LPI в Великобритании и Ирландии, поясняет, как сертификат влияет на карьеру в Linux.

Linux Format: Когда прошел первый экзамен LPI? **Билл Куинн [Bill Quinn]:** Официально LPI был зарегистрирован как некоммерческая организация в Нью-Брансуике, Канада, 25 октября 1999 года. И первый экзамен провели в том же году.

LXF: Важно ли получить LPIC, если вы намерены делать карьеру как инженер Linux? Это наилучший способ?

БК: Сертификация ИТ, включая LPI — крайне важный компонент резюме ИТ-профессионала. Он показывает наличие у вас базового уровня знаний, необходимого (в нашем случае) данной области промышленности. Если вы хотите делать карьеру в Linux, то LPI — главная мировая сертификация. На сегодня по всему миру проводится более 350 000 экзаменов и имеется более 125 000 сертифицированных профессионалов. Основное отличие программы LPI — в независимости от производителя: мы концентрируемся не на каком-то одном дистрибутиве Linux, а на технологии.

Есть масса способов сделать карьеру в Linux — всякие курсы, колледж, коммерческие курсы, самообразование или практическое обучение на работе. Но на некой ступени вашей карьеры вам неизбежно придется продемонстрировать базовый уровень знаний. Сертификация в признанной организации — самый простой способ для этого. Неважно, в какой форме человек обучался: ему всегда будет нужно показать свои знания посредством сертификации.

LXF: Говорят, наблюдается нехватка инженеров Linux. Вы согласны с этим? И если да, то что можно с этим сделать?

БК: В ИТ-индустрии наблюдается катастрофическая нехватка специалистов. В зависимости от системы оценки, полагают, что по этой причине к 2017 году в Европе будет от 300 000 до 1 000 000 вакантных мест в сфере ИТ.

Если сосредоточить внимание на развивающихся областях ИТ — мобильных технологиях (в частности, Android), виртуализации, облачных технологиях и больших данных, то все они основаны на Linux/Open Source. eSkills UK полагает, что к 2017 году будет нужно 69 000 специалистов в области больших данных. Спрос на профессионалов Linux велик, поэтому и уровень зарплат у них высок.

Великобритания осознает необходимость перемен, необходимость в знаниях компьютерных технологий. Это отражается в школьных реформах, от программы ICT до компьютерных технологий; но можно сделать больше. Нужно, чтобы колледжи и университеты осознали преимущества соответствия образовательных программ промышленной сертификации — и предлагали студентам и то, и другое. В Ирландии перемены происходят быстрее: правительственный План трудоустройства отражает потребность в большем количестве профессионалов Linux, чтобы удержать позиции технической столицы Европы.



LXF: По-вашему, даже опытные инженеры Linux увидят для себя преимущества, пройдя сертификацию LPI?

БК: Куда бóльшая проблема — молодое поколение. В индустрии ИТ наблюдается рост числа работников в возрасте 35–45 лет, а число 20–30-летних падает. Это потенциально приводит к долгосрочной проблеме, поскольку новое поколение в индустрию не спешит.

Однако все могут получить преимущества от программы непрерывного развития. В частности, в ИТ-индустрии нужно постоянно обновлять свои знания. С помощью многоуровневой программы сертификации, подобной LPI, сотрудники смогут расширить свои знания в более сложных областях. С LPI вам нужно проходить сертификацию каждый пять лет, так что у вас и у ваших сотрудников будут самые актуальные и самые необходимые знания.

Аарон Сакстон, директор по обучению UKFast



Linux Format: Как бы вы описали свою роль?

Аарон Сакстон [Aaron Saxton]: Мои обязанности как директора по обучению заключаются в выявлении и развитии талантов через образовательные программы, позволяющие пройти сертификацию международного уровня. Здесь в UKFast мы знаем, что расширение возможностей для правильных людей приведет не только к их индивидуальному росту, но и к росту компании, в которой они работают.

LXF: Насколько сертификация LPI важна для UKFast и для бизнеса в целом?

АС: Очень важна. Она дает нам возможность ознакомить наших инженеров с основами, не испортив их самопальными методами — я имею в виду те плохие привычки, которыми могли успеть обзавестись опытные инженеры. Например, одни методы могут быть более длительными, а другие — менее эффективными по сравнению с должными. Обучение на LPIC — отличный способ проверить способности и подтвердить умения, а также научиться тому, что пока не известно. Для новичков в Linux LPIC — то, что надо: нужная среда, контент и сертификация.

LXF: В UKFast разрабатывают свои обучающие материалы? Какие ресурсы вы бы посоветовали читателям LXF?

АС: Да, это не просто обобщенный подход — мы координируем цели и результаты обучения

с нашими целями бизнеса. Поскольку обучение LPIC легко сочетать с практической деятельностью, мы также предоставляем нашим молодым инженерам Linux [Junior Linux Engineers] виртуальные среды, настроенные так же, как некоторые решения для наших клиентов. Вашим читателям я бы порекомендовал CBT (www.cbttuggets.com), где предлагается обучающее онлайн-видео и практические экзамены. Обучающие руководства и электронные книги по LPIC — тоже отличный ресурс, и предлагают пробные экзамены и практические тесты. Я бы также посоветовал общаться с опытными инженерами, способными что-то посоветовать и подсказать. Это в природе Linux и сообщества открытого кода — быть открытыми и делиться; а тех, кто готов поделиться своими знаниями, вы найдете с помощью соцсетей и онлайн-форумов.

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.

Защита для высокоскоростных корпоративных сетей Ethernet 100/1000 Мбит/с

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

Основное средство защиты для реализации различных политик информационной безопасности с помощью:

- фильтрации пакетов на канальном, сетевом, транспортном и прикладном уровнях;
- управления транспортными соединениями между отдельными узлами ЛВС или виртуальной ЛВС (VLAN);
- контроля контента данных на прикладном уровне с учетом направления, времени и типа протоколов передачи трафика.

Дополнительное устройство защиты для:

- обеспечения безопасности функционирования ранее установленных в компьютерной сети средств защиты и устройств маршрутизации;
- мониторинга трафика с возможностью анализа данных регистрации пакетов по различным критериям и интеграции с IDS;
- обеспечения функционирования сетевых распределенных телематических приложений и GRID-ресурсов.

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center

Мир, каким его видит

Марк

Продираясь сквозь слухи и противоречия, Тэмсин Оксфорд выясняет у основателя Canonical, **Марка Шаттлворта**, что же происходит с Ubuntu.



Марку Шаттлворту (Mark Shuttleworth) порядком досталось от СМИ за последний год, когда множество аполетов Ubuntu, недоумевающих, куда он ведет платфор-

му, выступали с открытой критикой его действий. В этом, уже набравшем ход, году мы решили позадавать ему каверзные вопросы: узнать, над чем он сейчас работает, что им движет и примерял ли он когда-нибудь бикини в стиле принцессы Лей...

Linux Format: Давайте начнем с простого. В последнее время стало появляться всё больше ноутбуков на Ubuntu; самой известной маркой среди

них является Lenovo. Как вы считаете, эта тенденция продолжится в этом году, и нам стоит ожидать дальнейшего увеличения их числа?

Марк Шаттлворт: Мы наблюдаем, как среди производителей ПК растет понимание того, что сегодня им необходимо придерживаться мультиоперационной стратегии и что Ubuntu имеет значительный вес на некоторых рынках. На данный момент наша доля среди крупных брендов по всему миру составляет около 20% — естественно, речь идет о тех рынках, где Ubuntu используется широко либо является существенной частью стратегии правительства. Яркие примеры — Индия, Китай и Бразилия. Мы надеемся, что Ubuntu также укрепит свои позиции во Франции, Германии и других европейских странах.

LXF: Значит ли это, что Ubuntu пошла в народ?

МШ: Нет, я думаю, она по-прежнему для экспертов и знатоков. На Западе ее уж точно народной не назовешь. Она распространена в профессиональной среде — любой IT-специалист и тот, кто как-то связан с технологиями, сегодня почти наверняка использует Ubuntu. Так что в этих кругах она весьма популярна, но что касается конечных пользователей, то на Западе это лишь отдельные ценители. А вот на Востоке всё несколько иначе. С момента появления Ubuntu на рынках Индии, Пакистана, Китая мы отслеживаем показатели ее распространенности и наблюдаем активный рост. Это говорит о том, что людям удобнее пользоваться Ubuntu, установленной на ноутбуке, а не заменять ее чем-то другим.

LXF: А каковы, по-вашему, шансы, что, скажем, в Великобритании в последующие несколько лет пользователи изменят привычному выбору и обратят взоры в сторону Ubuntu?

МШ: Я знаю наверняка две вещи. Во-первых, среднестатистический пользователь сегодня выбирает множество технологий помимо Windows. В том числе и Интернет, и Mac OS, и iOS, и телефоны Android и iPhone, обнаруживая, что на всех этих платформах можно делать одно и то же. Здесь не возникает автоматически мысль, что для ПК может быть только одна платформа. Люди уже привыкли выбирать.

С другой стороны, мне кажется, что Ubuntu стала проникать и в сферы, не требующие специального знания. Вы можете установить ее вместо Mac или Windows, получив практически те же возможности. Если вы столкнетесь с проблемой,





Google вам в помощь. Вероятно, в целях популяризаторства пришлось бы много чего переработать, в том числе и внешний облик, более свободный по сравнению с традиционным ПК. Именно поэтому наша команда клиентской стороны концентрируется на оттачивании планшетных интерфейсов, это сейчас во главе угла.

Мы также работаем над широким спектром проектов по мобильным платформам. Следует весьма серьезно относиться к новым веяниям в мейнстриме, ведь здесь одно нововведение ведет за собой другое. Именно так в этой среде возникли облачные вычисления, верно? А через них люди пришли к Ubuntu. Ведь она и до этого была более чем полноценной, но не так-то просто заставить людей отказаться от того, к чему они привыкли. Так что с позиции пользователя, важно поймать новое и сделать это популярным.

LX: Опять же, по поводу народности, вот вы работали с Valve, рассчитываете ли вы на то, что геймерам проще будет привыкнуть к непростым устройством Ubuntu?

О РАСШИРЕНИИ АУДИТОРИИ

«Ubuntu стала проникать и в сферы, не требующие специального знания.»

МШ: По-моему, мы уже много что упростили в Ubuntu. Бытует уже куча историй, когда кто-то давал кому-то из родственников флэшку с Ubuntu, и тот, не будучи IT-специалистом, в итоге возвращался с благодарностью, что всё установилось и прекрасно работает. Мне кажется, именно Ubuntu, по сути, удалось сделать Linux проще, и без ущерба для производительности

и безопасности, которыми он так славен. И, по моему, именно мы, геймеры, всегда хотим выжать из техники как можно больше, ведь это доставляет нам особое удовольствие. Мы делаем крупную ставку как на геймеров, так и на разработчиков, и, вероятно, именно это убеждает их в надежности Ubuntu.

LXF: Вам, должно быть, непонятно, к чему такой акцент на сложности, но совсем недавно вы сами в блоге выказывали недовольство тем, что люди воспринимают Ubuntu как нечто

мудреное, во что сложно вникнуть. Вы можете это прокомментировать?

МШ: Думаю, есть люди, которых в Linux привлекает безопасность и производительность, и действительно хорошие программисты попадают именно в эту категорию. Ну и есть также мнение, что Linux — нечто экзотическое и сложное, служащее подтверждением вашего выдающегося »

ума. По-моему, это полная чушь. Умные люди с простыми вещами мучиться не станут, им нужна как раз платформа, пригодная для решения сложных задач. Именно поэтому я выступаю за простоту Linux. Не столько в расчете на не-технарей, сколько на настоящих технарей, занимающихся действительно серьезными и интересными делами. Именно поэтому я так рад, что Ubuntu в ходу у настоящих профи — по-настоящему гениальных людей, решающих очень сложные задачи. Благодаря Ubuntu им не приходится размениваться на мелочи — она проста в использовании и много где признана, что, на мой взгляд, очень, очень ценно. А снобистский подход в духе «Я же умный: никто не разбирается, а я разберусь», на самом деле, не особо конструктивен.

LXF: Как вы думаете, почему Mir вызвал у многих такую негативную реакцию?

МШ: Здесь много причин — в том числе, у некоторых с появлением Mir прибавится забот. Эту небольшую категорию составляют разработчики инструментария и в целом специалисты по работе с графикой и интерфейсами. Сообщество Linux довольно разнородно, и, с одной стороны, они понимают, что в этом его плюс, ведь возможность выбора порождает конкуренцию, что дает Linux преимущество по сравнению с монолитными проприетарными решениями. С другой стороны, они легко заводятся из-за политики. Мы понимали, что текущая работа над графическими серверами нового поколения ведется не совсем так, как надо, и одна небольшая команда в Canonical пыталась продвигать альтернативную стратегию.

И по-моему, то, что у них получилось, просто феноменально. Их решение компактнее, оно лучше прописано и применимо на широком диапазоне оборудования. Оно быстро нашло поддержку у разработчиков инструментария, производителей игр и так далее, что поможет им добиться

желаемых результатов. Нами громко возмущаются за этот прорыв, но если посмотреть, от кого исходит эта критика, вы обнаружите, что это наши же конкуренты. Именно они больше всего недовольны, а всё остальное уже эхо, когда, знаете, поговорить больше не о чем, и есть некий кусочек кода, не существующий на практике, и о котором можно сколько угодно рассуждать, что все и делают.

Что касается меня, я очень рад вкладывать деньги в ПО, и считаю это правильным. По-моему, конкуренция — штука хорошая, и мне кажется, мы уже привлекли на свою сторону довольно много людей, просто показав, как Mir работает и почему он создан именно так. Я испробовал Mir на множестве устройств, и он прекрасно себя проявил, делая работу более аккуратной, быстрой, слаженной. В общем, по-моему, иногда, ради важной цели, надо идти слегка наперекор. Мое самоутверждение здесь ни при чем. Дело только в коде. Мы могли бы использовать чужой, могли поручить его написание какой-нибудь из наших команд, но для меня важно, чтобы в конечном итоге за это взялись лучшие люди — и посмотрим, что у них получится. Я очень рад, что мы делаем Mir, и думаю, умным людям понятно, что его своеобразие и есть его сила, и это несколько не коснется конечных пользователей.

LXF: Какие, по-вашему, еще достоинства у Mir?

МШ: Думаю, главные я уже назвал, верно? Команда Mir реализовала возможности автоматизированного тестирования любого аспекта Mir. Без проверки в Mir ничего не добавляется. И меня просто поразило, когда некоторые из наших конкурентов, по другую сторону баррикад, попытались представить это как недостаток. Тоже мне ковбой. Запускать серьезное ПО без тестирования, без автоматизированной системы проверки качества, значит просто надеяться на авось, и по-моему, это полное безумие.

Мы знаем, как выглядит качественное ПО, и на мой взгляд, Mir подходит по всем статьям. Это код, который работает незаметно, но сбои в нем будут очевидны всем. Вы никогда не будете использовать Mir, но он лежит в основе всего. Поэтому меня поистине удивляет то, что разработка на основе автоматизированного тестирования стала едва ли не главным объектом критики. Это потрясающе, что наша команда выбрала такой путь, и по-моему, то, что публика готова, очертя голову, набрасываться на профессионалов, со всей серьезностью и осторожностью подходящих к написанию кода, говорит о некоторой незрелости сообщества Open Source.

Но я считаю, глупо всю жизнь бояться, что тебя осудят. Нужно браться и делать то, что считаешь нужным, и быть готовым убить пересмешника. А хулители пускай себе потом хватаются за голову.

LXF: А как насчет Trusty Tahr? Всё идет по плану? И как правильно произносить — Tahr или Tear?

МШ: Правильно — Tahr. Mir является единственным графическим движком, пригодным также

и для мобильных систем, и непосредственно сейчас мы работаем над конвергенцией. Сейчас у нас есть единый архив кода, и на его основе разрабатываются две модели. Одна — для мобильной Ubuntu, другая — для настольной. Они на 99% одинаковы, но кое-что их всё же отличает. Графическим сервером на мобильной платформе выступает Mir, в качестве инструментария настройки выступает X. То есть Mir находится в Trusty Tahr, но это касается только пользователей мобильных систем. Для настольных же предлагается использовать сам Mir.

Попробовать стоит. Думаю, любому, кто интересуется технологиями, стоит заглянуть под капот и посмотреть. Те, кто решился, уже своими глазами убедились, что это действительно прекрасная работа, аккуратно и добротнo выполненная. В Mir нет никакой коммерческой подоплеки, это не попытка подмять под себя всё ПО, это просто команда профессионалов, знающих свое дело и решивших сделать нечто быстрое, чистое и очень надежное в самой критической области, и очень здорово, что в Canonical их поддержали.

LXF: Вы упомянули, что любое сообщество завязано на политике — не этим ли, в том числе, объясняется ваш переход от роли сотрудника к роли скорее лидера? Есть ли на то другие причины?

МШ: Это неверное толкование того, о чем я говорил, или, возможно, я неудачно выразился. Я считаю, что в мире, где все мнения равны, ничего невозможно добиться. Поэтому первое, что следует, на мой взгляд, преодолеть сообществу Open Source — это как раз то, что все голоса имеют одинаковый вес. Так не бывает. Есть люди, обладающие более широкими знаниями, нежели другие, у каждого свои достоинства и недостатки, и если вы хотите добиться успеха, необходимо построить сообщество, где у всех будут разные навыки, ведь чтобы что-то получилось, нужно задействовать все возможности.

В Ubuntu мы добиваемся этого за счет открытости к сотрудничеству с людьми из любых сфер, с любыми компетенциями. У нас вклад переводчика ценится не меньше, чем вклад разработчика, но оборотная сторона заключается в том, что мы очень четко рассчитываем, кто за что должен отвечать. Вы же не поручите разработчику из Германии перевод на китайский, верно?

Наша цель, в плане организации усилий, в том, чтобы каждый всегда получал по заслугам. У нас работают нужные люди, делающие правильный выбор. Сегодня в мире Open Source есть своя идеология, и по ней, все мнения на форумах равноправны, но на практике, внутри Ubuntu мы стремимся осознанно выбирать тех, кому мы доверяем принимать ответственные решения и готовы им следовать. И такой подход уже себя оправдал. Именно так Ubuntu пришла к успеху.

Отчасти это означает, что когда вы доверяете кому-то сделать сложный выбор, результат потом может вас сильно удивить. И если вы его не принимаете, значит и доверия, по сути, нет. Есть один удачный пример. Много лет назад я попросил



разработчиков ядра и команду, занимающуюся безопасностью в Ubuntu, подобрать гипервизор для виртуальных машин, который будет впоследствии использоваться как основной. В те времена на пике популярности был *Xen*, его использовали все, это было самое стандартное решение среди гипервизоров в Linux, но команда, к моему удивлению, выбрала *KVM*. Совершенно неожиданно. И моя первая реакция была: «Нет, этот вариант я не поддерживаю, больно мне нужно оригинальничать». Но все наши лучшие специалисты были за, а они владели всей информацией и могли уйти куда угодно, на любую платформу. Они выбрали *KVM* без всякого личного интереса, и мы поддержали *KVM*. Со временем он стал использоваться в Ubuntu по умолчанию, а затем, спустя много лет, его подхватили и Red Hat, и все остальные.

По-моему, это прекрасный пример того, как важно принять вызов, убедиться, что ты нанял правильных людей, и они ответственны за свой выбор. Вы контролируете их действия и то, чем они руководствуются, но окончательный выбор оставляете за ними. Если во всем следовать за толпой, навсегда останешься посредственностью, принимая только то, что и так очевидно. Посмотрите, че-



платформами. А стоит Canonical что-то сделать самостоятельно, сразу раздаются возгласы, что мы не как все. Но они хотят, чтобы мы следовали не за сообществом, а за его конкурентами. Я полностью за реально открытый код: не хотите повторять то, что уже сделали другие — пожалуйста, используйте готовое, но двигайтесь вперед, создавайте что-то лучшее.

Мы вновь получили награду как лучший дистрибутив года, можете считать это популяр-

ризаторством, ведь присуждается она рядовыми пользователями Linux, но я считаю, что, вопреки всем злопыхателям, люди голосуют за то, чему они доверяют, и, так или иначе, это Ubuntu.

LXF: Насколько изменилась Ubuntu по сравнению с тем, какой вы изначально себе ее представляли?

МШ: А вот это интересно. Невозможно заниматься чем-то 10 лет, не впитав в себя много нового. Если рассуждать в духе «знай я тогда то, что знаю сейчас», что бы я изменил? Неизбежно найдется что-то, что удивит, или что поменялось. Что меня действительно радует — видеть, как сегодня и в сфере корпоративных, и клиентских систем, Ubuntu находится в центре всех самых важных сдвигов, таких как облачные вычисления.

Amazon, по большей части, работает на Ubuntu, равно как и практически всё, что базируется на облаке Microsoft и в частной облачной среде. Это самый большой и значительный прорыв в структуре корпоративных платформ после Web. Именно поэтому мне так нравится именно данная сфера: мы находимся в самом центре облачных вычислений и самой инфраструктуры, платформы как сервиса. Ее история началась с Ubuntu, и это неожиданно и прекрасно.

С клиентской стороны всё сложнее, потому что типичный настольный пользователь по-прежнему

предпочитает Windows или Mac, но самое интересное здесь, пожалуй, связано с конвергенцией. Мне кажется, мы уверенно лидируем благодаря тому, что приняли ряд сложных решений в связи с переходом на Unity. Именно это приносит мне самое большое удовольствие от работы. Делать что-то, что приведет к инновациям, или находиться в самом центре этого процесса.

LXF: Ну и простой вопрос — если бы вам довелось попасть на Kid's Hack Day (регулярное мероприятие для детей, интересующихся программированием, — прим. пер.), что бы придумал восьмилетний Марк?

МШ: Думаю, была бы пара глобальных задач на горизонте, в том числе, квантовое программирование. Современное вычисление черно-белое, включено — выключено, и есть куча всего интересного, что можно сделать с программами, лежащими в их основе. Используя только Вкл и Выкл, Да и Нет, вы можете заставить машины сотрудничать и выполнять задачи произвольно. Это создает иллюзию расплывчатых серых пространств внутри этой черно-белой вселенной. Квантовое программирование, по сути, как раз интересно такой расплывчатостью; думаю, восьмилетнего меня это заинтересовало бы.

Ну, а другая глобальная задача касалась бы пищи и энергии. Взгляните на нашу планету сейчас: вы увидите, что на ней осталось очень мало свободного места; а потом рассчитайте потребление калорий для человечества на следующие 50 лет. Где взять столько еды? Если вы едите мясо, то внезапно осознаете, что мы уже практически израсходовали все предельные участки земли, и теперь посягаем на всё остальное. Неплохо бы восьмилетним задуматься над решением этих проблем.

LXF: Может быть, вы скоро нас чем-нибудь порадуете? Чем можете сейчас поделиться?

МШ: Не люблю преждевременных откровений, но честно признаюсь, что никогда не носил бикини в стиле принцессы Леи. **LXF**

О ПРИНЯТИИ РЕШЕНИЙ

«Если во всём следовать за толпой, навсегда останешься посредственностью.»

го добилась Ubuntu, пройдя путь от одного из многих сотен дистрибутивов до всеобщего любимца, и, что еще важнее, пионера в абсолютно новых сферах, таких как преобразование сантиметров в пиксели [y-conversion]. Все остальные рабочие окружения Linux следуют за Ubuntu — правда, выражается это по-разному, и расценивают они это каждый по-своему, но факт тот, что все они значительно изменили свой подход к ПК после Ubuntu. И это, опять-таки, потому, что мы в принципе никогда не идем за толпой, мы ищем подходящих людей. Неважно, работают ли они на Canonical или просто являются членами сообщества, нам абсолютно всё равно, ведь нам нужны самые одаренные, и они должны быть у руля...

LXF: Внутри сообщества сложилось представление, что вы решили «работать на себя» и не особо стремитесь следовать за остальными, что мы уже обсудили. Однако, даже если правда за Canonical, что от этого выиграет сообщество?

МШ: Я просто скажу, что, в большинстве случаев, возгласы насчет «работать на себя» принадлежат нашим конкурентам. Они ведут какой-нибудь открытый проект, и стоит нам что-то возразить, они говорят: «Ну, конечно, вы же не идете за сообществом», но при этом предлагаемый ими проект тоже не принадлежит сообществу, а разработан Intel, не так ли, или другими конкурирующими

Как делают Pi

Нейл Мор дивится, как Sony удается штамповать Raspberry Pi миллионами.

История Raspberry Pi стала своего рода британским триумфом. Он начался с желания лучше обучить студентов программированию и электронике. Идея преподавателей и бизнесменов — британцев британцами же и воплощена в прототипе, который затем пошел в серийное производство в Технологическом центре Sony в Уэльсе, недалеко от Кардиффа.

После подробных интервью с вдохновителями Pi Эбеном Аптоном [Eben Upton] [LXF173, стр. 38] и Питом Ломасом [Pete Lomas] [LXF181, стр. 44],

мы решили разузнать, как собирают эти платы-бестселлеры. Как дешевая плата может конкурировать с китайскими аналогами во времена, когда британская промышленность считается почившей в бозе (даже несмотря на ее изготовление на заводе Sony в Великобритании)?

Чтобы узнать это, мы отправились в Пенкод, городок в Уэльсе в 14 км к западу от Кардиффа, который встречает огромным заводом Sony. Там Linux Format встретился с Питом Ломасом, доверенным лицом фонда Raspberry Pi,

и Гэретом Джонсом [Gareth Jones], старшим менеджером отдела развития предприятий Технологического центра Sony в Великобритании (UKTEC). Джонс рассказал, что в начале 2000-х они выпускали телевизоры Sony Trinitron, но после революции плоских панелей перешли на дорогое оборудование для телевидения и потребительскую электронику, в про-

Начнем с печатной платы Raspberry Pi, с которой нам помог Ломас, ее создатель. Печатная плата Pi — двусторонняя, с поверхностным монтажом; на верхней стороне платы размещены односторонние компоненты со сквозным монтажом. Проще говоря, компоненты размещены по сторонам платы, а некоторые проходят сквозь плату и припаиваются с противоположной стороны.

Все начинается с готовой пластины из шести готовых плат для Pi, которую Лиз Аптон [Liz Upton] окрестила «Pi'вной упаковкой» [“six-pack” — блок из шести банок пива, — прим.

«Идея преподавателей и бизнесменов — британцев британцами же и воплощена.»

изводстве которой завод сейчас лидирует среди всех предприятий группы Sony.

Ломас и Джонс любезно провели для нас экскурсию по предприятию и по процессу изготовления Raspberry Pi. Как объясняет Пит, «все просто — мы припаиваем компоненты на печатную плату, формируя механические и электрические соединения, тестируем ее и отгружаем». Но вопрос в том, как UKTEC удается выпускать более десяти тысяч Raspberry Pi в день, т. е. по штуке каждые 2,9 секунды?

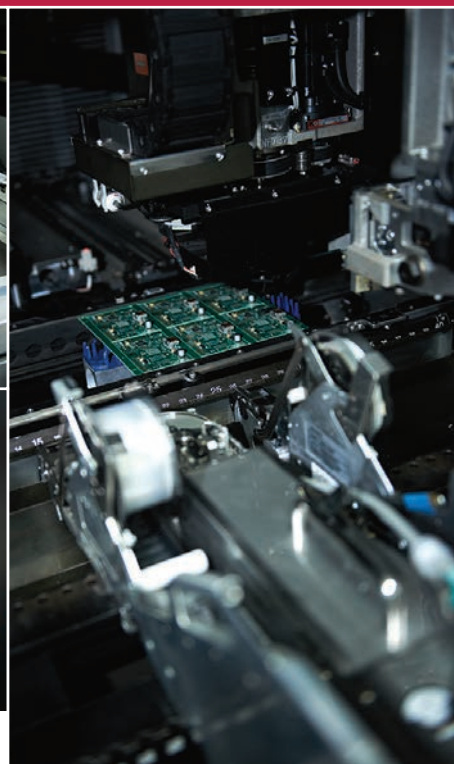
пер.]. Все производственные линии на предприятии Sony проектируются индивидуально для каждого проекта. Квадратная форма пластины позволяет надежно обрабатывать несколько Pi сразу. На пластине вокруг каждой платы есть припуск, что упрощает обработку плат во время производства, а также область для надежного зажима плат на месте: компоненты вставляются в платы со скоростью 5,5 штук в секунду, и необходимо исключить смещение и вибрации платы. Пит отметил, что часть компонентов



➤ Нанесение паяльной пасты по трафарету.



➤ Пример трафарета нанесения пасты (это не плата для Pi).





перевешивается через край платы, в помощь монтажу.

Процесс изготовления состоит из четырех этапов: монтаж компонентов на нижней стороне платы, монтаж компонентов на верхней стороне платы, монтаж сквозных компонентов и полное тестирование и упаковка каждой платы. Ломас отмечает, что компетенция Sony по понятным причинам не ограничивается только производством Pi, а охватывает и начальное и последующее приобретение компонентов и технологические процессы, необходимые для производства Pi. Проследим же за превращением Raspberry Pi из стыдливого обнаженной платы в готовое к бою и полностью протестированное устройство. Процесс мы рассмотрим для одной стороны платы — на самом деле он повторяется дважды.

Первый этап — покрытие печатной платы паяльной пастой. «Паста» на самом деле состоит из сфер диаметром около 25 мкм, перемешанных с флюсом. Наличие флюса важно: он позволяет пасте течь и защищает ее от окисления во время пайки, обеспечивая надежное соединение. Паста наносится через металлический трафарет толщиной всего 0,1 мм, и получаются дорожки диаметром 300 мкм. Олово может застрять в трафарете, поэтому каждая плата проходит оптическое тестирование на компьютере. Если что-то не так, можно удалить пасту и заново «перепечатать» плату.

Умные машины

После проверки дорожек можно добавить на плату компоненты с помощью машины для поверхностного монтажа — из собственных

Сложные компоненты

Не все компоненты изготавливаются одинаково. Ключевой аспект компетенции Sony UKTEC в производстве Pi — приобретение компонентов, насчет чего у Sony UKTEC есть строгие правила. Чтобы гарантировать использование только качественных компонентов, Sony тщательно оценивает производителя. На полную проверку счета на материалы для Raspberry Pi уходят месяцы.

Это позволяет избежать проблем, с которыми изначально столкнулись при производстве Pi, когда разъемы Ethernet заменили неудачно, и пришлось от них отказаться.

Компоненты должны быть не только подходящими, но и правильно упакованными. Технология автоматического поверхностного монтажа требует,

чтобы компоненты поставлялись на бобины, и чем больше бобина, тем лучше. Для простых компонентов их количество на бобине может достигать 10 000. Производственная линия Sony работает так быстро, что даже такие бобины приходится заменять раз в несколько часов. Для замены бобины приходится останавливать всю линию.

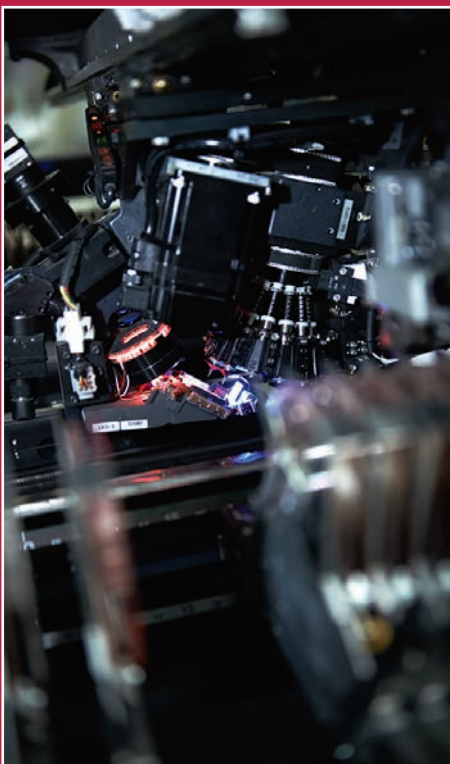
Эта система тоже умная. Все машины отслеживают использование компонентов и автоматически направляют запросы в хранилище. Умной система является потому, что новые бобины находятся, заказываются и доставляются для замены именно в тот момент, когда исходная бобина вот-вот закончится. Штрих-коды пустой и полной бобин сканируются, что исключает вероятность ошибок.

машин предприятия Sony UKTEC. Компоненты находятся в бобины или кармашках и высасываются оттуда вакуумным соплом, подключенным к механизму для монтажа. Эта умная машина позволяет быстро и точно монтировать компоненты с помощью матрицы датчиков. Размеры компонента могут составлять всего 1×0,5 мм, а допуск на размещение — всего лишь 40 мкм (толщина человеческого волоса) — вот и представьте себе требуемую точность. Расположение компонента в сопле проверяется камерой. Перед установкой механизм при необходимости поправляет компонент. На полной скорости эти машины могут размещать 25 000 компонентов в час.

У каждого прибора может быть до 12 вакуумных сопел и парных установочных головок,

работающих совместно друг с другом. Пока одна головка перемещается для установки одного компонента, другая возвращается за новым. Эти машины умеют устанавливать до 60 000 компонентов в час, но Pi с ее 173 поверхностно размещаемыми компонентами довольствуется более скромной машиной на 25 000 компонентов в час. Даже на ней «Pi'вную упаковку» можно полностью покрыть компонентами всего за 150 секунд.

Самые сообразительные из вас заметят нестыковку в числах. 150 секунд на размещение 173 компонентов на шести Raspberry Pi и скорость выпуска по одной Pi каждые 2,9 с не соответствуют друг другу. На самом деле, в эту цифру вносят вклад три машины: одна собирает нижнюю сторону платы и две — верхнюю,



» Автоматическая оптическая проверка припоя.

» Свежая «Pi'вная упаковка» перед запуском в производственный процесс.



причем для повышения эффективности производится балансировка нагрузки всей системы. Инженеры-технологи гарантируют, что каждый процесс в среднем всегда занимает одно и то же время. Если какой-то этап замедляется, система перехватывает управление и уменьшает скорость работы, а для решения проблемы подключается дополнительное оборудование.

Возможно, вы слышали о том, как некоторые пытаются починить свои приставки PlayStation и Xbox, положив их в духовку, чтобы припой расплавился и процессор можно было вернуть на место. С упаковками Pi происходит примерно то же самое, только в огромной промышленной печи под управлением компьютера. Сейчас компоненты просто лежат на этих маленьких шариках припоя. Достаточно нагреть шарики — и компоненты будут припаяны к плате. Печь состоит из нескольких зон, в каждой из которых температура повышается до того момента, когда припой расплавится и сформируется соединение. После этого происходит постепенное охлаждение печатной платы. Это немного посложнее, чем перевести регулятор вашей духовки в положение «8», но опять же иллюстрирует уровень компетентности инженеров-технологов UKTEC.

Здесь становится жарко

Нам нужно найти «золотую середину» — температуру, достаточную для расплавления припоя, при этом не слишком высокую

и не слишком низкую. Чтобы найти это «рабочее окно», нужно сжечь несколько Pi, поместив рядом с ними прибор для определения температуры, называемый «кротом [mole]». Этот процесс повторяется до тех пор, пока не будет найдена верная температура. И теперь плата Pi уже готова? Нет.

Плата проходит автоматическую оптическую проверку с помощью HD-камеры. Она сравнивает плату с эталоном, проверяя наличие, положение и соединение для каждого компонента и все остальное, что может выглядеть неправильно. При обнаружении любых ошибок оператор выполняет визуальный контроль платы. Для демонстрации работы опти-

«Команда проверки качества проверяет... качество проверки качества!»

ческой проверки Пит показывает бракованные микросхемы, у которых не было уголков.

Но оптической проверкой тестирование плат не ограничивается. Предприятие Sony UKTEC всеми силами старается исключить «возможности появления дефектов». При сборке Raspberry Pi таких возможностей сотни — дефект печати паяльной пасты на плате, отсутствующий компонент или некачественная спайка. Программы сортировки и тестирования гарантируют, что некачественные платы не покинут производственный участок. Даже по прохождении всех тестов Pi ждет команда проверки качества, которая фактически

проверяет качество проверки качества! До клиента ни должна дойти ни одна неисправность.

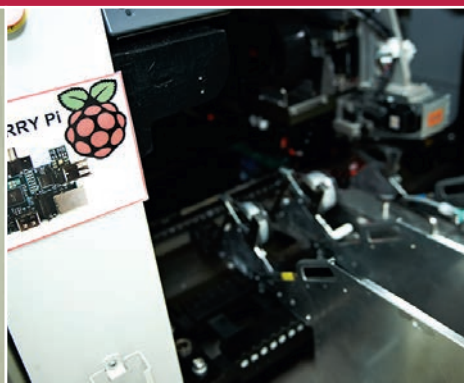
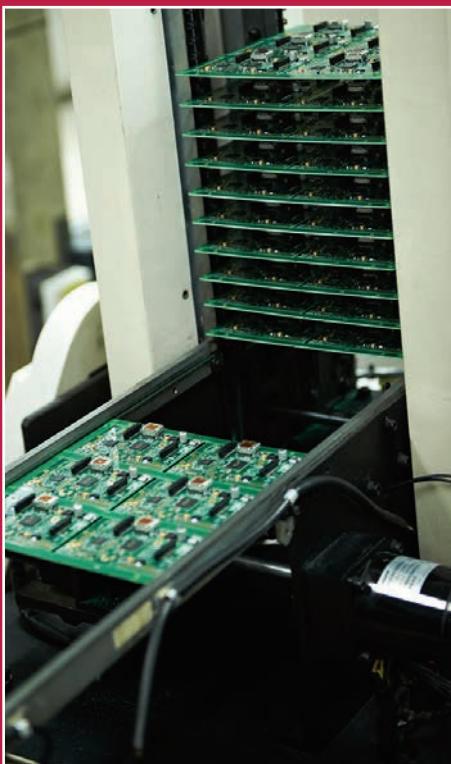
Мы хотели было сказать, что с нижней стороны платы все точно так же, но плата Pi обладает некоторыми интересными особенностями, из-за которых нижняя сторона платы чуть сложнее верхней. Ломас поместил на Pi интересный модуль памяти, который размещается на однокристалльной схеме. Для установки модуля памяти на BCM2835 он погружается в припой, а затем размещается на однокристалльной схеме Broadcom. Для погружения используется лоток, заполненный паяльной пастой, уровень которой поддерживается постоянным с помощью вращающегося скребка.

Затем эта сторона «пропекается», и модуль памяти размещается на ней, после чего производится дополнительная оптическая проверка.

В этот момент можно проверить качество соединений,

и так как любые найденные отклонения могут означать наличие ошибок в производственном процессе, их можно будет исправить в зародыше. Качество припайки памяти проверить трудно, но оптическая проверка по крайней мере проанализирует расположение краев схемы. Там, где проверить выравнивание оптически нельзя, для определения спаек и отсутствия шариков припоя у корпусов микросхем используется рентгеновское сканирование.

На этом этапе в плату вручную вставляются микросхемы со сквозным монтажом. Таких микросхем всего пять — например, это порт Ethernet, которому не повредит



► Проверка плат, сошедших с производственной линии.



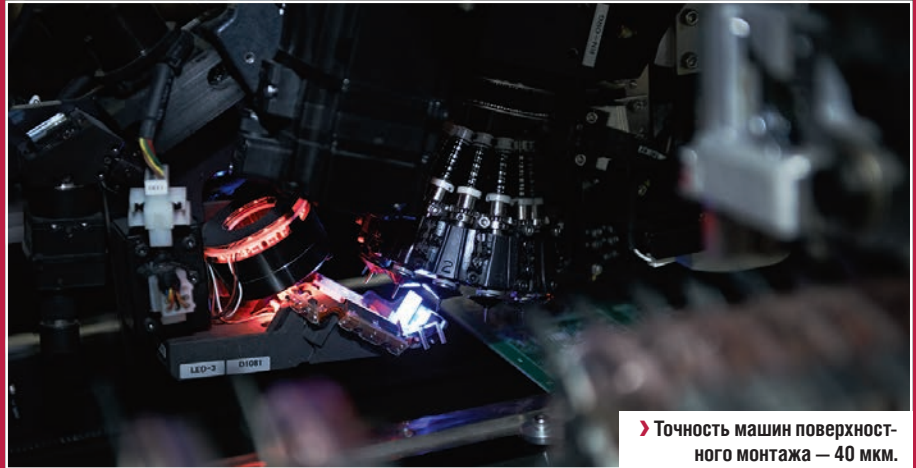
► Raspberry Pi помещают в теплонепроницаемую и непроницаемую для припоя кассету.

дополнительная механическая прочность. После этого пластина помещается в тепло- непроницаемую и непроницаемую для припоя кассету для процесса с устрашающим названием — пайки волной припоя.

Как следует из названия, это ванна с припоем, через которую протекает постоянная волна припоя, циркулирующая через плавильный котел. Для качественной пайки высота, форма и температура волны контролируются в соответствии со стандартами точности. Если волна будет слишком низкой, соединения не получится, если слишком высокой — плату затопит припоем. На самом деле, через смесь даже пропускаются пузырьки азота, которые уменьшают появление окислы, окисленного припоя.

При прохождении через волну припоя компоненты с поверхностным монтажом накрываются для защиты от повреждений и расплавления. Перед прохождением платы через волну припоя на нижнюю сторону платы напыляется флюс. Это защищает ее от окисления и гарантирует хорошее качество соединения. Платы предварительно нагреваются, чтобы после нанесения припоя не остыл слишком быстро, и это еще более улучшает качество соединения.

После этого «Pi'вная упаковка», готовая к полному тестированию, прибывает на упаковочную станцию. Каждая Pi извлекается из панели и помещается на модуль для тестирования. У каждой станции есть два модуля тестирования: на один можно помещать или убирать плату, а другой используется для выполнения тестов и записи в устройство различных заводских кодов, например, модели



› Точность машин поверхностного монтажа — 40 мкм.

и предприятия, на котором оно было изготовлено. Плата Raspberry Pi, успешно прошедшая проверки, помещается в антистатический пакет и в коробку — теперь она готова для отправки. Все бракованные платы исследуются на наличие проблем, и найденные проблемы анализируются, так как это может повлечь изменения процесса изготовления.

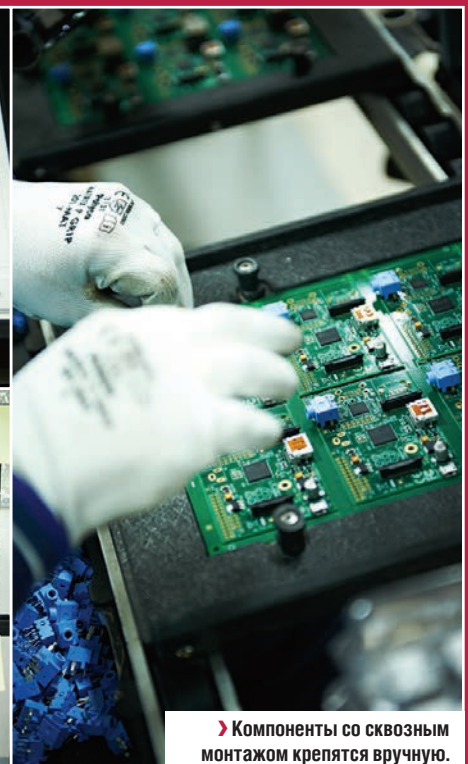
Рождение Pi

Ломас отмечает, что с таким количеством выпускаемых плат есть вероятность сцепления компонентов уголками, что скорее всего приведет к отбраковыванию платы. Фонд Raspberry Pi тесно взаимодействует с Sony для выявления и исправления таких проблем, и в рамках этого сотрудничества уже было внесено несколько изменений в схему печатной платы. Мы спросили у Sony, какова доля

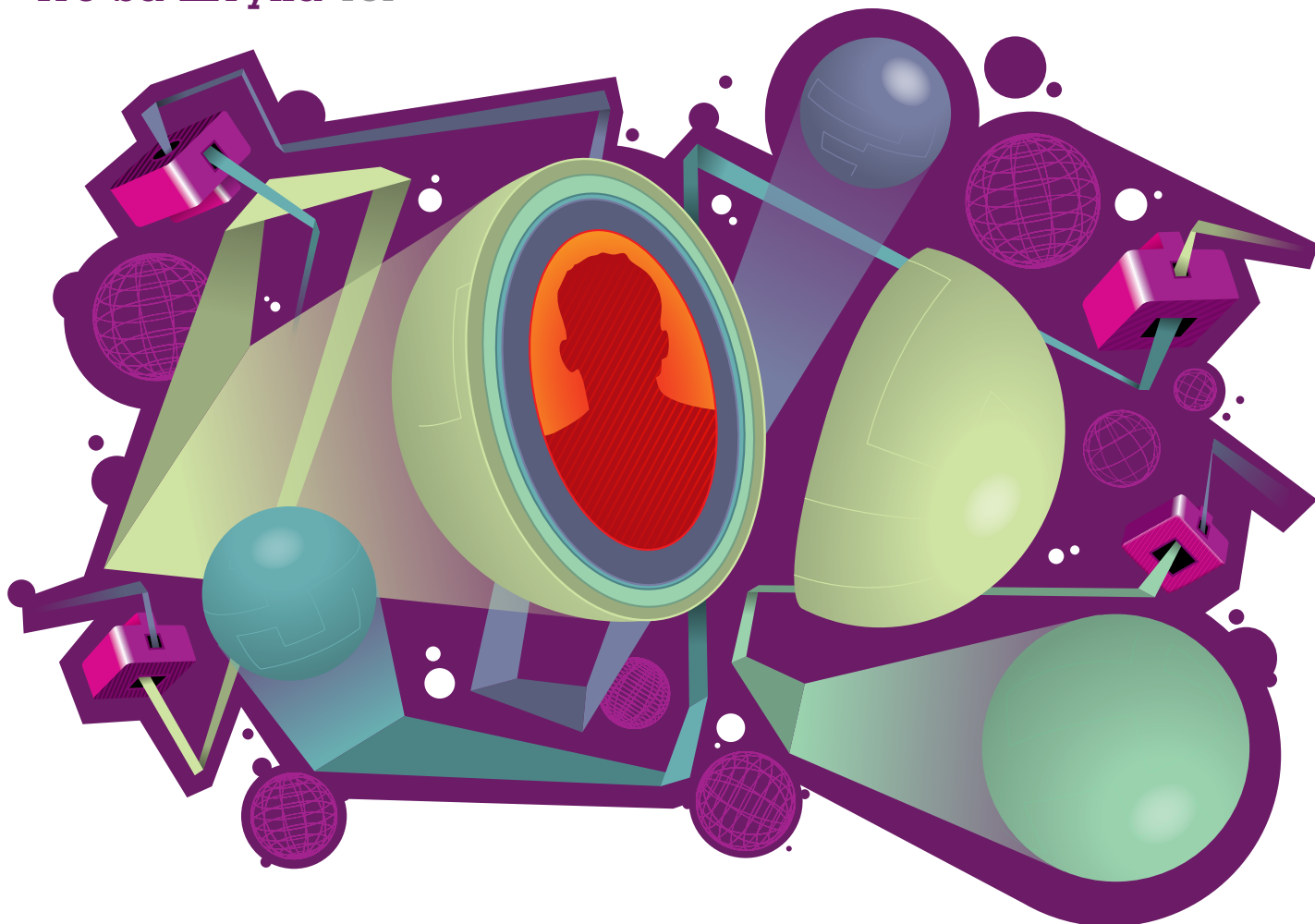
неисправных плат, и она оказалась такой маленькой, что ее трудно измерить. На заводе количество неисправных плат — несколько штук на миллион, а количество плат, возвращаемых клиентами для оценки неисправностей, оценить трудно. В мире компьютерной техники, где доля возврата в 1% считается хорошим показателем, доля возврата у Pi, по словам Ломаса, составляет 0,03%. Это выдающийся результат Sony. Особая тщательность, внимание и производственный опыт Sony при изготовлении Raspberry Pi — неотъемлемая часть истории успеха Pi. Они также гарантируют, что фонд Raspberry Pi всегда сможет удовлетворить большой спрос на Pi у пользователей. Мы получили прекрасные впечатления и не устаем благодарить команду Sony UKTEC за то, что они разрешили нам подорвать их производительность, пока мы делали снимки. LXF



› Готовые для тестирования Pi.



› Компоненты со сквозным монтажом крепятся вручную.



Что за штука... Tor?

Нам на электронную почту пришло письмо с описанием важного компонента сетевой анонимности. Видимо, от **Марко Фиоретти**, но кто ж его знает.

В Что же за штука *Tor* и зачем на его логотипе луковица?

О *Tor* (www.torproject.org) — это свободная программа, с которой бродить по Интернету анонимно стало значительно проще. До луковицы мы доберёмся немного погодя.

В А вот мне скрывать нечего. Я не преступник и не шпион, зачем мне *Tor*?

О Конечно, своей жизнью рисковал бы только Джеймс Бонд, если бы СПЕКТР перехватил его переговоры. А шпионить за всеми просто ради удовольствия — идея абсолютно нелепая, из разряда фантастики. Анонимность в сети важна по другим причинам. Она может защитить обычных людей от несправедливой ценовой политики, отказа в предоставлении услуг или чрезмерно навязчивой рекламы.

В Интересно. А конкретные примеры привести можете?

О Например, интернет-магазины могут завышать цену на товары и услуги, просто узнав,

кто вы и где находитесь, и что вы можете заплатить больше. Иногда они могут вообще отказать вам в обслуживании. Если вам в это не верится и кажется надуманным, вспомните о региональной блокировке DVD и о сайтах, воспроизводящих кино или музыку и недоступных для некоторых стран.

В Но где они берут информацию о местопребывании?

О Любой компьютер, подключённый к Интернету, имеет уникальный IP-адрес, имеющий существенную привязку как к местонахождению, так и пользователю и его провайдеру. Все пакеты, которые вы отправляете по Сети, содержат IP-адреса в полной форме, доступные как отправителю, так и получателю.

В И эта информация всё время находится в открытом доступе? А без нее никак?

О Конечно! Задумайтесь на секунду: не будь адреса постоянно доступны в нешифрованном виде, в стандартном формате и в специальном

месте (именуемом заголовком пакета), поделенными на ячейки, то каким образом все эти маршрутизаторы между вами и сайтом узнали бы, на какую именно страницу направить ваш запрос? А затем отправить эту страницу именно вам, а не кому-то другому?

В Ясно. Стало быть, эту важную информацию невозможно зашифровать без потери пакетов.

О Именно. Но во всех приведённых мной примерах «жертва» определяется не напрямую, а просто проверкой, какую систему она использует или хочет подключиться. Поэтому даже у обычных граждан могут быть веские причины защищать то, что они делают или говорят, а также то, откуда и куда идёт информация.

В Чем же в этой проблеме способен помочь *Tor*?

О *Tor* перенаправляет все ваши пакеты по крайней мере через три различных сервера *Tor*, прежде чем они отправятся в пункт назначения.

В И как же это сделает меня анонимом, если IP-адреса зашифровать невозможно?

О Начнём с того, что цепь серверов, именуемых на языке *Tor* узлами [node] или передаточными (транзитными) станциями [relay], выбираются случайно для каждого подключения и меняются, по умолчанию, каждые 10 минут.

Кроме того, говоря по-простому, клиент *Tor* шифрует данные и конечный адрес при помощи

В Вроде все то же самое, в чем тут разница с *Tor*?

О VPN даёт возможность пользоваться веб-сервисами, где важна скорость соединений, поскольку там нет промежуточных переадресаций, как в *Tor*. Кроме того, некоторые поставщики услуг и правительства принципиально блокируют все потенциальные узлы *Tor*, но не все VPN. Минус VPN в том, что вам нужно авторизоваться,

О Мы вновь упрощаем, но давайте представим себе некий сайт, который посещают только по воскресеньям, допустим, с полуночи до часу. Кто-нибудь сможет с уверенностью сказать, что вы туда заходили, если в это время в сети был только один ваш домашний компьютер. Сколь бы маловероятным это ни казалось, такого рода хронометраж является свидетельством, против которого *Tor* не особо эффективен.

«Систематическая попытка разрушить анонимность в *Tor* невыполнима.»

ключа, известного только последнему узлу. Клиент в зашифрованном виде передаёт данные, вместе с IP-адресом конечного узла, в другие пакеты, промежуточному узлу. Новая ячейка данных формируется с другим ключом, известным только этому промежуточному узлу. Затем...

В Понял! Результат будет зашифрован ключом отправителя и передан по назначению!

О Примерно так. Кроме того, *Tor* минимизирует время задержки обработки пакетов и гарантирует их целостность, но суть его в этой многослойной, по принципу луковицы, маршрутизации (*Tor* в исходном варианте было задумано как сокращение от The Onion Router — Луковичный маршрутизатор).

За исключением конечного узла, все прочие узлы в цепи *Tor* получают пакеты, расшифровать которые могут только они сами, а внутри пакетов содержатся данные, зашифрованные другим ключом и передаваемые на какой-то случайный компьютер. Поэтому узел не может выяснить, что в них содержится, куда они отправлены и откуда. Кроме того, даже если один из узлов выйдет из строя или пропадёт, безопасность системы не пострадает.

В Подождите-ка. Чем тогда *Tor* отличается от Virtual Private Networks?

О Целым рядом ключевых вещей. VPN создаёт прямые, полупостоянные туннели между удаленными компьютерами, заставляя их работать так, будто они подключены друг к другу непосредственно.

VPN часто служит для связи разных удаленных офисов внутри одной организации. Когда один компьютер в сети VPN выступает в Интернете как прокси, другие тоже могут его использовать, даже когда прямая маршрутизация заблокирована или отслеживается.

чтобы её использовать, или самим создавать свою собственную. *Tor*, напротив, полностью децентрализован: его невозможно полностью закрыть, и вам не нужно спрашивать ни у кого разрешения.

В Обеспечит ли *Tor* полную анонимность в Сети?

О Конечно, нет! *Tor* необходим, но им одним не обойдётся. Вот если бы вы были детективом, а я делал бы пересадку в метро на каждой остановке, чтобы скрыть, куда я направляюсь, могли бы вы вычислить, когда я окажусь в определённом месте?

В Я мог бы подложить передатчик вам в карман или поставить своих людей на каждой станции, чтобы они мне сообщили, как только вы появитесь...

О Именно! *Tor* уязвим для всех цифровых аналогов подобных стратегий. Например, cookies вашего браузера — это всё, что нужно сайту, чтобы вас узнать. Кроме того, если какая-нибудь программа на вашем компьютере отправит ваш реальный IP третьим лицам, *Tor* будет бессилён.

В Верно. А как насчет тех «промежуточных станций», которые *Tor* использует?

О Промежуточные узлы *Tor*, через которые проходит ваш трафик на пути к пункту назначения, можно отслеживать. На самом деле, было бы очень и очень странно, если бы некоторые правительства до сих пор не завели сами такие узлы, в качестве ловушек. Выход в том, чтобы всегда использовать *Tor* поверх HTTPS. Это полностью зашифрует ваш обмен данными с теми сайтами, которые вы посещаете.

В Есть ли что-нибудь, с чем *Tor* не сможет справиться?

В Насколько же можно доверять свою конфиденциальность *Tor*?

О Согласно нескольким независимым и надёжным источникам, код *Tor* достаточно надёжен. Любая систематическая попытка разрушить анонимность трафика *Tor* по всему миру практически невыполнима, даже для супер-секретного агента любого правительства с горами денег...

В Расскажите побольше об этих узлах: легально ли завести себе такое?

О Теоретически — да, хотя юридически это ничем не подкреплено. За подробностями обращайтесь к LegalFAQ на сайте *Tor* (<http://bit.ly/16IA54i>), особенно если вы думаете завести конечный узел.

В Почему? В чём разница между этими двумя типами?

О Почти наверняка можно утверждать, что рано или поздно через каждый конечный узел *Tor* в Интернет пройдёт запрещённый трафик. Узел будет зафиксирован в качестве его источника, и его администратор будет первым, кто попадёт под пристальное внимание полиции.

В Стало быть, промежуточные узлы гораздо безопаснее, так?

О Да... с учётом всех вышеупомянутых подводных! Помимо того, что их труднее поймать на передаче запрещённого контента, промежуточные узлы могут оправдать то, что они не могут знать, что именно пересылают.

В Верно, я готов! С чего лучше начать знакомство с *Tor*?

О Попробуйте *Tor Browser Bundle* на официальном сайте (www.torproject.org). Это защищённая версия *Firefox*, работающая на Linux, Windows и Mac со съёмного носителя и по умолчанию использующая *Tor* и HTTPS, но без JavaScript и плагинов. Хотя лучше, конечно, взять безопасный дистрибутив Linux [см. Анонимные дистрибутивы, LXF174, стр. 26], куда, помимо этой сборки, войдут и другие безопасные свободные программы. Для телефонов Android мы бы порекомендовали прокси *Tor*, под названием *Orbot*. LXF



По рецептам доктора Брауна

Д-р Крис Браун

Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

Фокус со Шляпой?

Уж не знаю, что выйдет из так называемого «объединения сил» CentOS и Red Hat, и судя по многим комментариям участников сообщества, я не один такой. Я даже не знаю, как это назвать. Это и вправду партнерство, или поглощение?

Похоже, для CentOS все меняется очень быстро. Уже обновлен сайт, и четыре ведущих разработчика CentOS перешли в Red Hat, хотя «заслонка» между Red Hat и CentOS, по-видимому, останется, и CentOS будет по-прежнему зависеть от появления RPM с исходным кодом, как зависит и сейчас.

В Red Hat говорят о преимуществах CentOS как о стабильной платформе, которой разработчики могут пользоваться для тестирования, и о разработке новой «инфраструктуры сборки», способной упростить выпуск «вариантов» CentOS, наверное, по аналогии с Debian. Но ведь той технологией, которую обкатывали для Red Hat, была Fedora. Она что, уже не работает? CentOS всегда казался мне свободным «клоном» Red Hat для тех, кто не хочет платить за поддержку (и на несколько месяцев отстает). Поэтому я полагал, что Red Hat с ним смирилась, хотя это и украденный у них бизнес.

Стивен О'Грэйди [Stephen O'Grady] с Red Monk пишет: «...этот шаг Red Hat — логичный прием соседней экосистемы в распростертые объятия». Но таковыми можно и удуть, а «соседняя экосистема» уж слишком напоминает «конкурента» на языке Дилберта [американская серия комиксов о жизни компаний, — прим. пер.].
chris.linuxformat@gmail.com



Утилита i-nex

Удобный инструмент, собирающий в одном пункте всю труднодоступную информацию об устройствах и ОС.

Чтобы собрать информацию об устройствах своей системы, придется-таки покопаться. В этом вам помогут команды **lshw** и **lspci** и утилиты вроде **df** и **fdisk**. Также есть файл **/var/log/dmmsg** и все файлы в **/proc** и **/sys**, но если вы похожи на меня, то разобраться в **/sys** вряд ли сумеете.

Так или иначе, в этом месяце мне попалась графическая утилита **i-nex**, которая собирает всю эту информацию в одном месте. Я сказал, графическая? Ну, отчасти. Она управляется мышью,

но содержит в основном текстовые поля. На отдельных вкладках отображается информация о процессоре, графическом сопроцессоре, материнской плате, звуковых устройствах,



i-nex собирает разнообразную информацию об устройствах и системе всего на 10 вкладках.

дисках, версиях системного ПО, версии ядра, использовании памяти, сети и USB-устройствах. Информация представлена ясно и компактно.

Одна странноватая особенность на вкладке с процессором — ряд маленьких кнопок рядом с каждой характеристикой процессора (например, FPU или PAE), по нажатию на которые открывается страница из Википедии с описанием этой возможности. В программе также есть кнопка для создания снимка окна **i-nex**, которая немного удобнее, чем отдельная утилита для снятия скриншотов.

Но, пожалуй, самая нужная кнопка — та, что генерирует текстовый отчет с информацией на вкладках и выводом ряда команд, с помощью которых была получена эта информация. Она даже автоматически загрузит информацию на сервис Pastebin — это очень удобно, если вы хотите получить какую-то помощь по своей системе.

Я установил последнюю версию **i-nex** в Ubuntu 12.04 безо всяких проблем, хотя нужно включить PPA для разработчиков **i-nex** и **gambas-3**, от которых зависит **i-nex**, например, так:

```
sudo add-apt-repository ppa:nemh/gambas3
sudo apt-add-repository
ppa:i-nexdevelopment-team/daily
sudo apt-get update
sudo apt-get install i-nex
```

(Я не несу ответственности за эти команды: я их просто стянул с домашней страницы **i-nex** (<http://i-nex.linux.pl>). Также можно загрузить Deb-пакеты с Sourceforge (<http://bit.ly/1cwGdfd>). Разработчики, похоже, склонны к Ubuntu; подозреваю, что установить **i-nex** в системе на основе Red Hat будет сложнее.

Что такое Gambas?

Gambas — язык программирования и среда разработки, основанная на интерпретаторе BASIC. Да, на BASIC — с объектно-ориентированными расширениями (пример не уникален: VB.NET от Microsoft тоже является полностью объектно-ориентированным).

У Gambas прекрасная среда разработки (также написанная на Gambas) и быстрый компилятор, формирующий файлы классов, которые затем интерпретируются (как в Java или .NET). Но это необычная зависимость для низкоуровневых утилит вроде **i-nex**.



МОДЕРНИЗАЦИЯ
РЕГИОНАЛЬНОЙ ЭКОНОМИКИ
И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

24-25 АПРЕЛЯ

ИНФОФОРУМ



ВЛАДИВОСТОК
Дальневосточный
федеральный
университет

Дальний ВОСТОК

V-я Межрегиональная конференция
по региональной и информационной
безопасности

**ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ
СОВРЕМЕННОГО ПРЕДПРИЯТИЯ**

ИНТЕРНЕТ И БЕЗОПАСНОСТЬ

**НАЦИОНАЛЬНАЯ ПЛАТЕЖНАЯ СИСТЕМА
И ТРАНСГРАНИЧНОЕ ВЗАИМОДЕЙСТВИЕ**

**ТЕХНОЛОГИИ БЕЗОПАСНОГО ГОРОДА
И БЕЗОПАСНОСТЬ НА ТРАНСПОРТЕ**

dfo.infoforum.ru

Проникнем в самое сердце

Ядро — одна из важнейших, хотя и самых незаметных частей системы. Но чем оно, собственно, занимается? Добрый Доктор объясняет.

Ядро — это система, на которой работает Linux. Вообще-то пуристы заявили бы, что название “Linux” относится только к ядру и ни к чему другому. Они скажут вам, что все остальное — оболочка, утилиты командной строки, рабочие столы и т. д. — просто пользовательские приложения. Но мне кажется, что большинство читателей имеют более широкие взгляды: по сути, многие пользователи Linux не видят ничего, кроме браузера и, возможно, почтового клиента, и даже не знают, что есть такая штука, как ядро.

Да, но что оно делает?

Итак, что же именно делает ядро? Ну, основных функций у ядра насчитывается шесть:

» **Управление памятью** Каждой программе, которая работает в Linux, нужна память. Ядро отвечает за отведение этой памяти и за то, чтобы память каждой программы была отделена от памяти других программ. Если нужно выделить большой объем памяти, ядро будет сбрасывать области памяти на диск и возвращать их обратно в оперативную память при обращении к ним. Это называется «подкачкой страниц», и для этого в большинстве дистрибутивов Linux выделяется раздел подкачки.

» **Планирование процессов** Запущенным программам нужно процессорное время, и еще одна задача ядра — выделять время каждой из программ. Количество фактически выполняемых программ в любой момент времени не может превышать количество ядер вашего процессора. Но если «отступить назад» и посмотреть

на это в большем масштабе времени, то покажется, что все программы выполняются одновременно.

Если я спрошу вас, что вы делали сегодня утром, вы вряд ли скажете, что все время вдыхали и выдыхали, хотя на самом деле не делали ничего важнее этого. Точно так же, управление памятью и планирование процессов — жизненно необходимые, но самые незаметные снаружи функции ядра.

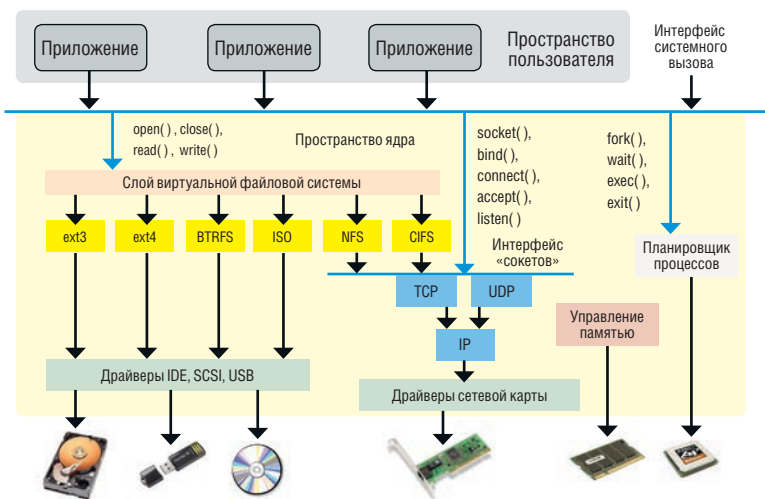
» **Файловая система** Файловая система — одна из самых заметных функций ядра. Мы все знаем, что такое файлы и каталоги, и привыкли с ними работать. Конечно, файл — это лишь иллюзия, создаваемая ядром (здесь прекрасно подходит термин «абстракция»); на самом деле данные разлиты по секторам на жестком диске, флэшке или удаленном файловом сервере. Существует множество типов файловых систем — ext3, ext4, Btrfs и т. д. Тем не менее, ядро предоставляет нам постоянную, понятную абстракцию для работы со всеми типами файловых систем с помощью полудюжины системных вызовов. (Мы поговорим о системных вызовах через минуту.)

» **Сетевой стек** Каждый раз, когда вы запрашиваете страницу сайта, происходит довольно сложный обмен датаграммами между вашим компьютером и сервером. Это работает благодаря тайному сговору различных протоколов, включающих TCP, UDP, IP и ICMP. Сетевой стек в ядре отвечает за решение этой задачи.

» **Драйверы устройств** Эти драйверы и программы на самом деле понимают, как работают устройства — например, как выбрать и прочитать сектор на жестком диске, как установить скорость сетевого интерфейса и т. д. Драйверы реализованы в виде модулей ядра, подробнее о которых мы поговорим чуть позже.

» **Управление доступом** Многим из нас знакомы права доступа к файлам `rwX`, но реализует их в конечном счете ядро. Без управления доступом ни о какой безопасности не было бы и речи. В некоторых ядрах с помощью добавочных модулей, реализующих SELinux или AppArmor, безопасность усилена дополнительно.

» Очень упрощенная схема некоторых крупных компонентов ядра и некоторых системных вызовов для доступа к ним.



Чего в ядре нет

Мнения о том, что стоит и не стоит включать в ядро, различны. Наверное, в ядро нельзя не включить какое-то управление памятью, планирование процессов и межпроцессное взаимодействие. Но другие вещи, такие как файловые системы и сетевые стеки, можно прекрасно реализовать в «пространстве пользователя» — это так называемый подход «микроядра». В Linux это не так, и технически ядро Linux называется «монолитным». (Подробнее см. врезку «Великие дебаты».) Но все согласятся, что такие вещи, как тригонометрические функции, обработка строк и вычисление регулярных выражений, не должны решаться ядром.

Великие дебаты

В 1992 году Энди Таненбаум [Andy Tanenbaum] (профессор по вычислительным дисциплинам в Амстердамском свободном университете) опубликовал в новостной группе `comp.os.minix` отчасти провокационный комментарий «Linux не нужен»,

утверждая, что микроядра превосходят монолитные ядра (учтите, что это было в 1992 году).

Линус Торвалдс не замедлил ответить на комментарий, и разговор перетек в то, что многие посчитали «войнушкой», хотя оба участника потом

упорно доказывали, что никакой личной вражды между ними не существовало. Этот разговор (довольно резкий) во всей его красе можно найти на <http://bit.ly/1kXaL35>, а второй раунд битвы двух «Т» — на <http://bit.ly/1cwIB5C>.

И кто же автор ядра?

Это непростой вопрос. Изначально все ядро писал Линус Торвалдс, но теперь он не входит даже в тридцатку ведущих разработчиков. Код обогащается немалым количеством разработчиков без явной

принадлежности к какой-либо компании. Но с точки зрения компаний, пять ведущих (по количеству отправленных заплаток, в порядке убывания) — это Red Hat, Intel, Texas Instruments, Linaro и SUSE.

Подробности можно найти в отчете о разработке ядра Linux Foundation. Также можно зайти на <http://kernelhub.org> и оценить активность сообщества по числу ежедневно отправляемых заплаток.

Архитектура ядра

Хотя ядро Linux иногда называют «монолитным» (чтобы отличить его от микроядра), на самом деле оно модульное. В нем есть основная часть, загружаемая при запуске системы, но большая часть функциональности обеспечивается модулями, загружаемыми динамически по запросу. При настройке нового ядра вы должны решить, какие компоненты войдут в основную часть ядра, какие будут скомпилированы в модули, и без каких вообще можно обойтись.

Преимущество такой схемы в экономии оперативной памяти. Создатель дистрибутива может включить в ядро исчерпывающую поддержку устройств (драйвера устройств обычно реализуются в модулях), не раздувая образ ядра. Есть преимущество и для разработчиков ядра: они могут упаковать экспериментальный код в модули, которые во время тестирования можно загружать и выгружать, не перезагружая систему. Основная часть ядра находится в каталоге `/boot`. Ее имя будет примерно таким: `vmlinuz-3.8.0-34-generic`. Удалять этот файл было бы опрометчиво! Если у вас установлено несколько версий ядра, то будет и несколько файлов с подобными именами.

Модули представляют собой скомпилированные двоичные файлы с расширением `.ko` (предполагаю, что это означает «kernel object» — объект ядра) и находятся в каталоге `/lib/modules/xxx/kernel`, где «xxx» — ваша версия ядра. Номер версии ядра легко определить, запустив команду

```
uname -r
3.8.0-34-generic
```

В моем случае модули помещены в каталоге `/lib/modules/3.8.0-34-generic/kernel`, и я могу вывести их список командой

```
ls -lR /lib/modules/3.8.0-34-generic/kernel
```

Опять же, если установлено несколько версий ядер, то в `/lib/modules` будет несколько подкаталогов. Список модулей также можно получить командой `modprobe -l`:

```
modprobe -l
kernel/fs/reiserfs/reiserfs.ko
kernel/fs/ext2/ext2.ko
kernel/fs/cramfs/cramfs.ko
kernel/fs/minix/minix.ko
kernel/fs/ocfs2/dlm/ocfs2_dlm.ko
kernel/fs/btrfs/btrfs.ko
kernel/fs/f2fs/f2fs.ko
kernel/fs/ceph/ceph.ko
```

Я опустил большую часть вывода (в нем 3700 строк), оставив лишь несколько с драйверами файловой системы. При этом загружена будет лишь малая часть этих модулей, которые можно просмотреть командой `lsmod`:

```
lsmod
Module Size Used by
vesafb 13876 1
rfcomm 47864 0
bnep 18258 2
bluetooth 247165 10 rfcomm,bnep
snd_intel8x0 38573 2
```

Опять же, я сократил список, хотя в нем было всего 34 строки. Здесь вы видите название и размер каждого модуля, количество

процессов, его использующих, и зависимые от него модули. Если модуль не используется, можно удалить его вручную (для этого нужны права root):

```
sudo modprobe -r rfcomm
```

Однако у системы достаточный инстинкт самосохранения, чтобы не удалять модуль, который в работе:

```
sudo modprobe -r bluetooth
FATAL: Module bluetooth is in use.
```

Также можно загрузить модуль вручную, например:

```
sudo modprobe btrfs
```

Впрочем, чаще всего ядро будет автоматически определять устройства и загружать соответствующие модули, так что вам эти команды могут никогда не потребоваться.

Может, вы помните историю о слоне и слепых. Один слепой трогает бивень и считает, что слон — это копьё. Другой находит ногу и думает, что это дерево. Третий ловит кончик хвоста и считает, что это метелка, и т.д. Смысл этой истории, как мне кажется, в том, что наш взгляд на любой предмет зависит от того, в каком контакте мы с ним находимся. Так и с ядром, на которое мы сейчас взглянем глазами трех наблюдателей: программиста, системного администратора и промышленного эксперта.

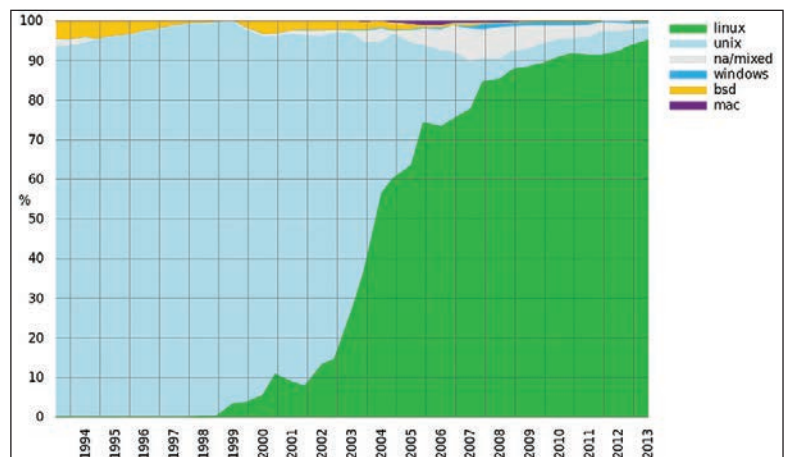
Взгляд программера

Наш первый наблюдатель, программист, скажет вам, что видимая часть ядра — это набор «системных вызовов», дающих доступ к функциям ядра другим программам. Системный вызов — это тщательно контролируемая точка входа в ядро, и по последним подсчетам, их было около 400, хотя некоторые из них устарели, а некоторые настолько туманны, что вы вряд ли когда-либо ими воспользуетесь. Если у вас установлены man-страницы, выполните команды

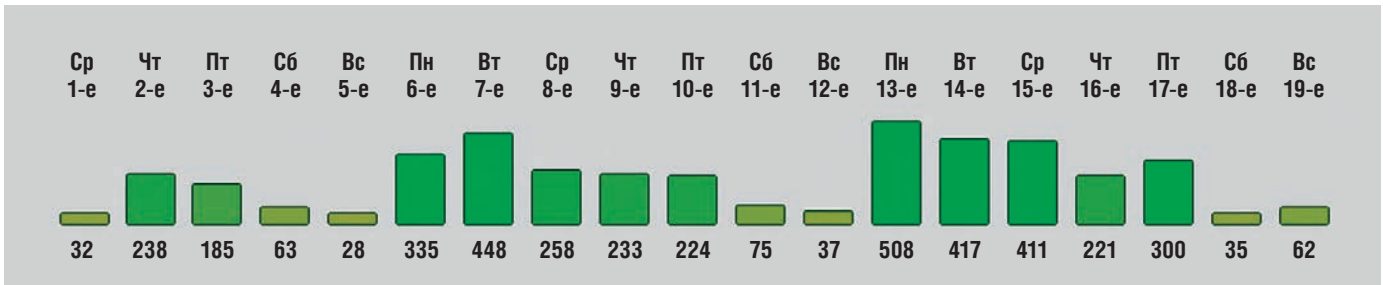
```
man 2 intro
man 2 syscalls
```

для получения подробного описания и списка команд. Чтобы понять, на что похож системный вызов, вот маленькая программка на C, которая копирует файл `foo`:

»



» Вглядываться незачем — большая зеленая область показывает взлет и доминирование Linux на рынке суперкомпьютеров.



➤ **Диаграмма с kernelhub.org** про количество заплаток в ядро за день. Даже у разработчиков ядра бывают выходные — по большей части.

```
/* Просто копирование файлов, но с системными вызовами */  
#include <fcntl.h>  
#define BSIZE 1024  
void main()  
{  
    int fin, fout; /* обработчики ввода и вывода */  
    char buf[BSIZE];  
    int count;  
    fin = open("foo", O_RDONLY);  
    fout = open("bar", O_WRONLY | O_CREAT, 0644);  
    while ((count = read(fin, buf, BSIZE)) > 0)  
        write(fout, buf, count);  
    close(fin);  
    close(fout);  
}
```

Если вам незнаком язык C, неважно. Обратите внимание на вызовы `open()` для получения «дескрипторов» входного и выходного файлов, на вызовы `read()` и `write()`, которые перемещают данные между файлами и буфером программы, и на вызов `close()`, который делает «уборку» в конце программы.

Это системные вызовы. Как видите, системный вызов ничем не отличается от любой другой библиотечной процедуры C. (На самом деле, то, что мы называем библиотечными процедурами — лишь обертки для выполнения команды-ловушки, необходимой для входа в режим ядра.)

При выполнении системного вызова происходит четкое переключение в более привилегированный режим, часто называемый «режимом ядра». Когда системный вызов завершается и управление передается программе, мы возвращаемся в «режим пользователя». Поэтому мы часто говорим, что код выполняется «в пространстве ядра» или «в пространстве пользователя».

Взгляд сисадмина

У нашего второго наблюдателя, системного администратора, отношения с ядром довольно нерегулярные. Он наверняка не вспоминает о нем каждый день, но время от времени устанавливает новую версию ядра или даже настраивает и собирает собственное ядро из исходного кода.

Во время загрузки он может выбирать, какое ядро загрузить. Но непосредственно с ядром он взаимодействует относительно

редко. Как мы упоминали ранее, он может вывести список модулей ядра, загрузить или выгрузить их, хотя ему вряд ли это понадобится. Он также может настроить некоторые параметры ядра путем изменения файлов в фиктивной файловой системе `/proc/sys` — например, включить маршрутизацию IP на компьютере, следующим образом:

```
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

или так:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Команда `sudo sysctl -a` выведет все параметры в `/proc/sys`, которые (потенциально) можно изменить таким способом, но я не удержусь от своего обычного предупреждения о том, что менять настройки можно только если а) вы знаете, что делаете, и б) вы как-то можете измерить изменение производительности системы вследствие изменения параметра. Подробности см. на map-страницах `sysctl` и `sysctl.conf`.

Взгляд эксперта

Наш третий герой, промышленный эксперт, наверное, опишет ядро как социальный и программный феномен. Он отметит высокую интенсивность совместной работы при разработке ядра. Согласно последнему отчету Linux Foundation, с 2005 года свой вклад в ядро внесли почти 10000 индивидуальных разработчиков из 1000 различных компаний — больше, чем в любом другом открытом проекте.

Скорость роста ядра феноменальна. В первой версии, выпущенной в декабре 1991 года, было около 10000 строк кода. В версии 3.10 чуть меньше 17000000 строк. (Конечно, при моей любви к компактности меня отчасти смущает такой большой объем.) В среднем каждый день в ядро принимается около 170 изменений, и новый релиз выходит раз в 60–80 дней.

Еще одним поразительным свойством ядра является впечатляющий набор оборудования, на котором оно работает. Linux портировали почти на все, что шевелится. Большинству из нас, наверное, кажется, что Linux есть по большей части на домашних компьютерах на Intel или AMD, ну или на ноутбуках, но на самом деле именно здесь его успех в достижении мирового господства был наименьшим.

Чтобы реально оценить проникновение Linux на рынок, нужно взглянуть на него шире. С одного края шкалы, Linux почти полностью доминирует в мире суперкомпьютеров, работая на 96% от 500 самых мощных суперкомпьютеров мира (<http://top500.org>). Linux даже занимает лидирующие позиции в облаках (см. комментарии Марка Шаттлворта [Mark Shuttleworth] в нашем интервью на стр. 46).

На другом конце шкалы вы найдете встроенный Linux — в широкополосных роутерах и других сетевых устройствах, системах развлечения пассажиров в самолетах, телевизорах Smart TV и абонентских приставках, а также (в виде Android) более чем на миллиарде устройств Android. Заглянув в посудомоечную машину или в холодильник, вы можете обнаружить его и там. Но, как однажды сказал Линус Торвалдс, «Вообще-то я не собираюсь уничтожать Microsoft. Это совершенно непреднамеренный побочный эффект». **LXF**

Хотите узнать больше?

Чтобы узнать больше о системных вызовах, наберите `man 2 intro` и `man 2 syscalls`; у каждого из системных вызовов также есть собственная map-страница в разделе 2. Если хотите узнать больше о внутренностях ядра, советуем попробовать <http://kernelnewbies.org>. Сайт предназначен для воспитания новых разработчиков ядра, и несмотря на слово «новичок»

в названии, он довольно быстро углубляется в довольно тонкие детали! Кроме того, на сайте Linux Foundation есть несколько прекрасных публикаций (<http://bit.ly/1cUzECD>), включая отчет о разработке ядра. Если хотите узнать, что появится в будущих релизах, взгляните на прогноз погоды Linux по адресу <http://bit.ly/1kCvulc>.

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Гонка технологий

Хороша та технология, которая наделяет вещи неожиданно ценными свойствами.
Георгий Александров

В принципе, эта проблема не нова, и человечество регулярно вынуждено выбирать между привычным и надёжным старым способом (лошади) и пугающим своей необычностью новым (паровозы). Но уж больно часто приходится сталкиваться с этим выбором в случае ПО. Только привык к чему-то — и нá тебе! — твоя привычка устарела, и надо окунуться в новый пользовательский опыт, опять... Глобальная Сеть позволяет относительно малой группе людей быстро растиражировать удачный опыт или прилипчивый мем на широкую аудиторию за вполне обозримый срок.

Скорость распространения новых технологий позволяет сменять их при небольших, а то и кажущихся преимуществах над старыми. Паровоз сменил лошадь потому, что на порядки ее превосходил. Новые же и активно продвигаемые технологии улучшения сжатия изображений превосходят прежние решения лишь на 10–20% — при том, что терабайтные диски давно уже типичное сегодня, а терабитный Интернет — не столь отдалённое завтра.

Возможно, хотя бы на время укротят не очень осмысленную гонку проекты вроде *mozjpeg*, недавно анонсированного Mozilla. *Mozjpeg* нацелен на оптимизацию всем привычной технологии сжатия на те же 10–20%, но без нарушения совместимости и смены привычек. Посмотрим, что из этого выйдет.
e.m.baldin@inp.nsk.su

В этом месяце вы научитесь...



Смотреть ТВ 62
Хочется смотреть футбол на Netflix, но претит иметь дело с *Silverlight*? **Нейл Ботвик** спешит на помощь, прихватив плагин *Pipelight*.



Собирать Gentoo 64
Джо Джордан объясняет, что такое дистрибутив Gentoo и с чем его едят. Мало того, еще и поможет его установить.



Не ходить в библиотеку ... 68
Станьте своим же электронным библиотекарем с *Calibre*. **Марко Фиоретти** (тайком, через SSH) скажет, как организовать ваши книги в коллекцию.



Качать торренты 72
Чтобы старый ПК зря не пылился в углу, **Маянк Шарма** гоняет на нем загрузку торрентов с *Deluge 7/24*. А хотите — делайте то же на Raspberry Pi.

Код в учебниках

Строки исходного кода помещаются в цветные плашки. Если строка кода не умещается в колонке, ее остаток переходит на следующую строку, в той же плашке:

```
procedure TfrmTextEditor.  
  mniWordWrapClick
```

А плашки разделены зазорами:

```
begin  
  mniWordWrap.Checked := false
```

АКАДЕМИЯ КОДИНГА



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

В этом номере

PHP
Овладеваем искусством обдирания информации с сайтов под руководством **Кента Ельчука** с. 76

OCaml
Гибкий и функциональный язык, производный от ML, представляет **Джульетта Кемп** с. 80

Lua
Джульетта Кемп занимается со скриптовым языком — столь же гибким, сколь и легковесным с. 84

Squeak
Введение в свободную версию языка Smalltalk, опять от **Джульетты Кемп** с. 88

Erlang
Андрей Ушаков взялся за общечеловеческие истины: тестирование программ с. 92

Плагин Pipelight

Не посмотреть футбол, потому что некоторые сервисы транслируют его только через *Silverlight* от Microsoft? У **Нейла Ботвика** есть дельное предложение.



Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ничем не расскажет, где находится центральный сервер.

В прошлом месяце в разделе Ответы был вопрос о том, как смотреть BT Sport в браузере Linux. На самом деле, это более широкая проблема, актуальная и в отношении таких известных сервисов, как *Netflix* и *Lovefilm*. Эти сайты используют *Silverlight* для применения DRM-защиты содержимого, разрешая просматривать видео, но не скачивать. *Silverlight* — проприетарная технология Microsoft, и хотя существует открытый аналог, *Moonlight*, он уже порядком устарел, а поставщики DRM-контента предпочитают работать с актуальной версией. Отвечая на этот вопрос, мы предложили решение с использованием пакета *netflix-desktop* для Ubuntu, но вскользь упомянули, что есть и альтернативный вариант, подходящий для браузеров Linux. А также отметили, что этот подход, *Pipelight*, возможно, заслуживает отдельного рассмотрения в учебниках грядущего номера *Linux Format*... вот и он.

Wine необходимо

Оба решения, как *netflix-desktop*, так и *Pipelight*, используют *Wine*. Пока что для просмотра такого типа файлов без установки в браузер плагина *Silverlight* для Windows или MacOS не обойтись. Это значит, что стойким приверженцам свободного ПО можно дальше не читать, ведь мы будем обсуждать практический способ применить для этого пусть и по минимуму, но проприетарное ПО.

Решение *netflix-desktop* появилось раньше. В нём, с помощью подлатанной версии *Wine*, можно запустить в Linux Windows-версию *Firefox*, со встроенным плагином *Silverlight*. Это важный шаг вперед, но не лишенный недостатков. Если вы пользователь Ubuntu, то для вас главным недостатком будет то, что браузер придется использовать только под *Wine*, и вам не удастся совместить это с выполнением других задач в Сети; и он также заберет повышенную долю системных ресурсов. Кроме того, решение работает только в *Firefox*. Если же вы используете не Ubuntu или

один из ее дериватов, самым существенным минусом для вас будет то, что пакет *netflix-desktop* доступен только в PPA Ubuntu.

Pipelight использует другой подход, хоть и заимствующий многое из *netflix-desktop*, вслед за ним применяя заплатку *Wine*. Вместо отдельного браузера, *Pipelight* запускает только плагин, служащий как бы интерфейсом плагинов Windows — именно он запускается через *Wine*. Это не только более эффективно, но также порадует и пользователей других браузеров. *Pipelight* доступен в виде пакетов в целом ряде дистрибутивов, или же можно собрать последнюю версию из исходников. Домашняя страница — на сайте команды FDS Team (<http://bit.ly/1d4ABZa>).

Установка Pipelight

Pipelight, в виде заплатки к *Wine* и ее зависимостей, ныне доступен в большинстве дистрибутивов. Пользователям Ubuntu необходимо добавить PPA *Pipelight*:

```
sudo add-apt-repository ppa:piplight/stable
sudo apt-get update
```

Если у вас уже стоит старая версия *Pipelight* или репозитории *netflix-desktop*, все это нужно предварительно удалить во избежание конфликта версий:

```
sudo add-apt-repository --remove ppa:michael/pipelight
sudo add-apt-repository --remove ppa:ehoover/compholio
```

а затем установить с помощью

```
sudo apt-get install pipelight-multi
```

Пользователям openSUSE тоже легко это сделать, командами

```
zypper ar --refresh http://download.opensuse.org/repositories/home:/rbos:/pipelight/<openSUSE version>/home:rbos:piplight.repo
```

```
zypper ref
```

```
zypper install pipelight
```

Тем, у кого стоит Fedora, придется немного потрудиться, установив сперва стандартные шрифты Microsoft:

```
sudo yum install cabextract
sudo rpm -i http://sourceforge.net/projects/mscorefonts2/files/rpms/msttcore-fonts-installer-2.6-1.noarch.rpm
```

затем добавить репозиторий и установить сам плагин:

```
sudo wget http://download.opensuse.org/repositories/home:/DarkPlayer:/Pipelight/Fedora_20/home:DarkPlayer:Pipelight.repo -O /etc/yum.repos.d/pipelight.repo
sudo yum install pipelight
```

Пользователи Arch могут установить *Pipelight* напрямую из AUR с помощью

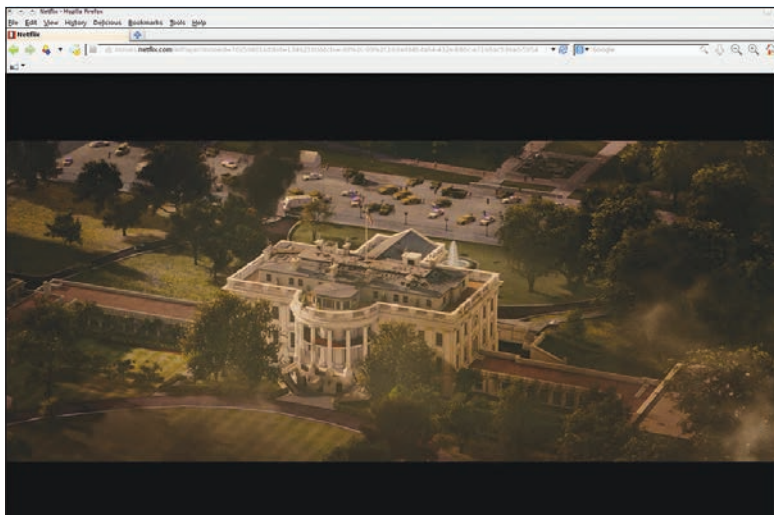
```
yaourt -S pipelight
```

А пользователи Gentoo — через оверлей *Pipelight*:

```
layman -a pipelight
emerge -a pipelight
```

Придется добавить пару пакетов в `/etc/portage/package.accept_keywords`, как это описано на домашней странице <https://github.com/ryao/pipelight-overlay>.

Пользователи других дистрибутивов, включая SteamOS, могут обратиться за инструкциями на сайт FDS Team (<http://bit.ly/1lq4XC3>).



» **Netflix** во всей красе: наслаждайтесь второсортным голливудским блокбастером в первоклассном браузере, на самой лучшей операционной системе!

Наконец, можно установить всё из исходников. Это задача нетривиальная, потому как прежде чем даже задуматься об установке *Pipelight*, понадобится специально залатанная версия *Wine*. Есть заплатки, изначально разработанные для проекта *netflix-project*, позволяющие, в том числе, плагину *Silverlight* творить свою чёрную DRM-магию — затем он и нужен многим сайтам. Пакеты дистрибутивов идут в комплекте с *Wine* и устанавливают его в качестве одной из зависимостей *Pipelight*.

Загрузка плагина

После установки необходимо активировать плагины Windows, которые вы хотите использовать. Да, плагины, во множественном числе, ведь *Pipelight* также поддерживает плагин Flash для Windows и прочие. Сейчас это не актуально, пока у нас есть родные Flash-плагины, но учитывая планы Adobe вскоре отказаться от их поддержки, на будущее пригодится. Если вы устанавливаете исходный код, выполните сначала

```
sudo pipelight-plugin --create-mozilla-plugins
```

В пакетных установках об этом уже позаботились за вас, и можно сразу приступить к настройке индивидуальных плагинов. Выбор команд для инициализации плагинов определяется тем, хотите ли вы установить их для всей системы или только для отдельного пользователя. Чтобы установить *Silverlight*, наберите

```
pipelight-plugin --enable silverlight
```

После этого начнётся скачивание и установка плагина Windows *Silverlight* в вашу домашнюю папку. Запустив эту же команду от имени root или используя **sudo**, вы установите плагин для всех пользователей. Просмотреть доступные плагины можно командой

```
pipelight-plugin --list-enabled
```

Отключить их можно как по одному, так и все сразу, используя

```
pipelight-plugin --disable silverlight
```

```
pipelight-plugin --disable-all
```

Теперь можно запустить браузер — первый раз неплохо бы сделать это из терминала. *Pipelight* завершит установку *Silverlight*, скачав всё, что ему нужно дополнительно, и запуск из терминала позволит тут же отследить, не возникло ли ошибок. Может появиться зловещее «Произошел крах плагина», когда вы попытаетесь его использовать. На некоторых машинах при первой загрузке так и было, но помогло нажатие кнопки перезапуска, и далее подобной проблемы не возникало. Набрав в адресной строке браузера **about:plugins**, вы увидите список установленных плагинов, с добавленным *Silverlight*. Сюда, равно как и на вывод при запуске браузера в терминале, определенно стоит заглянуть, если он работает не так, как надо. Если у вас действительно возникли проблемы, попробуйте другой браузер, чтобы определить, касаются ли они только браузера или всей установки *Pipelight*.

После установки *Pipelight* ваш браузер всё равно будет определяться как ПО Linux, и многие сайты это распознают, так что вам придется менять и идентификационную строку вашего браузера [User-Agent string]. Игнорирование этого действия может повлечь массу различных ошибок. А сделать это проще всего, установив подходящее для вашего браузера расширение, наподобие *user-agent-override* или *uacontrol* для *Firefox*. В первом есть ряд предустановок, из которых можно выбирать, тогда как второй позволяет использовать стандартные идентификационные строки для каждого сайта в отдельности. Для *Netflix*, попробуйте вот это: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:22.0) Gecko/20100101 Firefox/22.0

В *Chrome* и *Chromium* должно быть установлено расширение *user-agent-switcher-for-chrome*. Используя расширение, изменяющее идентификационную строку для отдельных сайтов, вы можете оставаться пользователем Linux для всех остальных. Только не пытайтесь выдать себя за *Internet Explorer*, поскольку тогда некоторые сайты попытаются заставить вас использовать

ActiveX, раз у вас *IE*. Не все средства для подмены User-Agent эффективны, но те, что мы упомянули, действительно работают, и если *Pipelight* не дружит с тем, что у вас, попробуйте один из этих вариантов.

Сообщалось также о проблемах при использовании любого переключателя с *Chromium* в Ubuntu 13.10. Но это частный случай: мы успешно опробовали *Chromium* на другом дистрибутиве, а на Ubuntu не возникло проблем с *Chrome*, так что выбор у вас есть. Ну или можно использовать *Firefox*; хотя одна из причин предпочтень *Pipelight* против *netflix-desktop* — как раз отсутствие привязки к *Firefox*. Ну, зато вы будете работать в *Firefox* для Linux, пока это не исправят.

Другие плагины

Хотя первоначальной задачей *Pipelight* было обеспечение возможности использования плагина Windows *Silverlight*, он применим и для других. Список поддерживаемых и экспериментальных плагинов можно просмотреть, введя

```
pipelight-plugin --help
```

и добавить любой из них с помощью опции **--enable**, подобно тому, как мы это сделали с *Silverlight*. Можно, с его помощью, использовать Windows *Flash*, что будет становиться всё актуальнее по мере старения версии для Linux. Вы также можете установить отдельные, более ранние версии *Silverlight*, если вдруг наткнетесь на сайт, где это понадобится. Некоторые утверждали, что сайт BT Sport лучше идёт на *Silverlight 5.0*. Возможно, что для некоторых сервисов проще всего будет воспользоваться небольшим скриптом. Например:

```
#!/bin/sh
```

```
pipelight-plugin --disable silverlight5.1
```

```
pipelight-plugin --enable silverlight5.0 firefox sport.bt.com
```

```
pipelight-plugin --disable silverlight5.0
```

```
pipelight-plugin --enable silverlight5.1
```

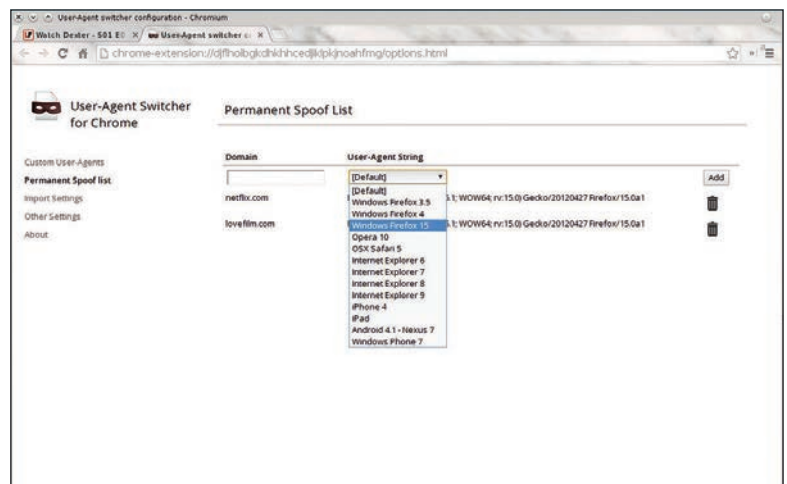
Создайте для него ярлык на рабочем столе и запускайте свои любимые фильмы одним щелчком! Можно также блокировать и разблокировать плагины, выполняя *pipelight-plugin* от имени root. Например,

```
sudo pipelight-plugin --lock-plugin shockwave
```

запретит любому пользователю запускать *Shockwave*. В зависимости от типа установки и пакета, экспериментальные плагины могут быть заблокированы по умолчанию, а потому вам придется использовать

```
sudo pipelight-plugin --unlock-plugin plugin-name
```

для того, чтобы их опробовать. Приятного просмотра ранее недоступных зрелищ! **!XPS**



➤ Расширение *Chrome User-Agent Switcher* поможет обмануть весь Интернет или некоторые сайты, предложив множество вариантов.

Gentoo: Ставим!

Ядро... а Gentoo тут при чем? Мы попросили **Джо Джордана** мягко провести вас по процессу установки дистрибутива с высокой степенью настройки.



Наш эксперт

Джо Джордан — разработчик программ, и на отдыхе от процеживания Сети в поисках жемчужин открытого кода он воскрешает старое оборудование новыми ОС Linux.

Что отличает Gentoo от других дистрибутивов? Настраиваемость. Во многих дистрибутивах есть *кое-какие* ограничения: типа, раздел `root` *должен* форматироваться в `ext4`, или в качестве загрузчика *надо* использовать `Grub2`, или еще что-нибудь. Конечно, если вам такое неважно, то вы ничего и не заметите; программа установки ОС настроит для вас все, и все вроде бы будет работать.

Gentoo — полная противоположность этой философии. Для начинающих: вы делаете полную установку вручную из командной строки, полностью контролируя настройку каждой части системы. Далее, менеджер пакетов работает совершенно не так, как в дистрибутивах на основе пакетов RPM или Deb. В Gentoo вы компилируете все требуемые программы из исходника. Тут, естественно, есть плюсы и минусы: с одной стороны, система станет невероятно быстрой; с другой — весьма медленный процесс установки, вплоть до того, что с обновлением системы ваша машина будет париться всю ночь. Процесс установки справедливо сравнить с марафоном: работа тяжелая, но в конце — награда! Контролируя все до мельчайших подробностей, вы многое узнаете о работе дистрибутивов Linux, превратившись в эксперта на случай сбоя или необходимости создать простую минималистскую систему. Стоит предупредить, что ваша первая установка продлится целый день, особенно если вы решите использовать среду рабочего стола.

Предупреждение: мы должны отметить, что данное руководство является дополнением, но не заменой `Gentoo Handbook` — справочника, помещенного на www.gentoo.org/doc/en/handbook. Рекомендуем вам установить Gentoo, используя его и наш урок; мы не будем повторять всю важную информацию из справочника! Вы предупреждены!

Как и во всех системах Linux, все начинается с загрузки ISO. Установите CD в дисковод, загрузите машину (мы использовали диск x86 на старом Pentium 4) и будьте начеку. Во-первых, нужно нажать на `Enter` до истечения 15 секунд после появления меню, иначе компьютер загрузится с жесткого диска; и во-вторых, вам предложат выбрать раскладку клавиатуры: американцам повезло, но всем остальным надо успеть очень быстро ввести код своей страны (для Великобритании = 40). По завершении загрузочной последовательности перед вами возникнет терминал `root live CD`; устраивайтесь поудобнее, мы в нем засидимся!

Но прежде чем что-либо устанавливать, скоренько убедитесь, что вы видите Интернет. Если на вашем компьютере есть порт Ethernet, рекомендуем на протяжении процесса установки

использовать его. А если у вас новый компьютер, которому «не нужно» проводное сетевое соединение, то все ответы найдутся в справочнике. Когда вы решите, что ваша сеть должна работать, попробуйте отправить пинг: `ping 192.168.1.1`, и нажмите `Ctrl+C`, когда получите ответные пакеты.

Готовим разбиение на разделы

Для начала заново поделим на разделы, или по крайней мере переформатируем, некоторые части жесткого диска или дисков. Выбрав раскладку разделов, вы готовы модифицировать диск. На нашем ПК мы использовали старый добрый `fdisk`, который берет адрес `/dev` нужного нам диска как аргумент: `fdisk /dev/sda`.

Если вы не уверены, какой диск будет каким, можете открыть `fdisk` на одном диске и ввести команду `p` — она покажет, что там есть. Если это не тот диск, о чем можно судить по списку разделов, выйдите командой `q` (с потерей всех изменений). Обратите внимание: получив сообщение о GTP/EFI, вы должны использовать `parted` вместо `fdisk`; загляните в справочник, чтобы узнать, какие команды использовать вместо приведенных.

Обнаружив нужный диск, можете использовать `m`, чтобы увидеть, какие команды доступны. Особо важны `d` для удаления раздела, `n` для нового раздела, `t` для изменения типа раздела (что вам придется сделать на разделе подкачки `[swap]`, если он у вас есть) и `w` для записи изменений в таблицу разделов. Затем `fdisk` запросит у вас дополнительную информацию, нужную для завершения команды; и помните, что он не позволит вам изменить размер существующих разделов (а `parted` — позволит).

Вы заметите, что команду `p` нужно все время запускать заново, поскольку журнал терминала нельзя «прокрутить назад». Причина в том, что этот низкоуровневый терминал забывает результаты, выходящие за пределы страницы. Есть два простых способа для команд обойти эту проблему: во-первых, можно добавить к команде `l less`, и она покажет вам результаты в пейджере `less`, а его можно прокрутить вверх и вниз; или добавьте `> log.txt`, чтобы сохранить результаты в файле (а `>> log.txt` заставит дополнение добавляться в существующий файл, не стирая старые).

Далее следует отформатировать свои разделы, сделав их пригодными к использованию. Команды у каждой файловой системы свои: например, `mkfs.ext2`, `mkreiserfs`, `mkfs.xfs` и `mkswap`. Каждая команда требует в качестве аргумента имя раздела (например, `/dev/sda1`), и если будет медлить, то покажет вам, что она делает. Последняя задача на стадии деления на разделы — смонтировать их, позволив начать создавать и редактировать файлы. Файловая система `live CD` имеет папку для монтирования разделов в `/mnt/gentoo`:

```
mount /dev/sda3 /mnt/gentoo
mkdir /mnt/gentoo/boot
mount /dev/sda1 /mnt/gentoo/boot
swapon /dev/sda2
```

Очевидно, что надо настроить имена разделов, которые вы только что создали и отформатировали, и если вы решите использовать отдельный раздел `/home` или `/usr`, создайте точку монтирования (папку) и смонтируйте соответствующие разделы.

```
.. Scanning for sda3b_generic... sda3b_generic loaded.
.. Scanning for xts... gf128mul, xts loaded.
>> Loading keymaps
Please select a keymap from the following list by typing in the appropriate name or number. Hit Enter for the default "us/41" US English keymap.
1 azerty  8 croat  15 fi  22 jp  29 pt  36 slovene  43 sf
2 be      9 cz      16 fr  23 la  30 ro  37 trf
3 bg      10 de     17 gr  24 lt  31 ru  38 --
4 br-a    11 dk     18 hu  25 mk  32 sc  39 ua
5 br-l    12 duorak 19 il  26 nl  33 sg  40 uk
6 by      13 es     20 is  27 no  34 sk-y 41 us
7 cf      14 et     21 it  28 pl  35 sk-z 42 wangbe
<< Load keymap (Enter for default):
```

И финальная проверка, перед тем, как записывать данные на диск: убедимся, что системное время установлено правильно. Для этого воспользуемся функцией **date**, которая должна выдать вам разумную строку (возможно, в UTC). Если время отличается более чем на несколько минут, неплохо было бы его настроить, таким образом: **date MMDDhhmmYYYY** (где буквы обозначают месяц, дату, час, минуты, и год соответственно). Тогда у нас не будет странных ошибок с файлами, созданными до 1970 года или где-то в будущем!

Скачайте Gentoo (как, опять?!)

Теперь мы готовы скачать и распаковать tar-архив структуры директории Gentoo по умолчанию и ряд простых бинарников команд (вроде **ls** и прочих низкоуровневых команд Unix), который в силу исторических причин именуется «tar-архив 3-го этапа». Для этого нам нужно убедиться, что мы находимся в файловой системе с чтением и записью, которая будет сохранена, если система даст сбой, т.е. нам надо сделать **cd /mnt/gentoo**.

Далее мы используем первую пока что не-терминальную программу под названием *Links*. Она работает в терминале «один экран за раз», но уже создает ощущение GUI, потому что поддерживает мышь (курсор — это большой толстый квадрат). Чтобы скачать tar-архив, мы направим *links* на официальный список зеркал Gentoo:

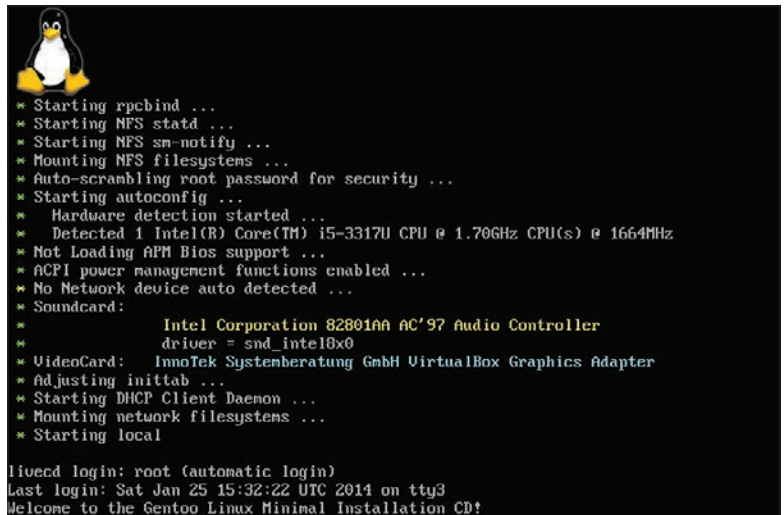
```
links http://www.gentoo.org/main/en/mirrors.xml.
```

Это страница с большим списком зеркал по всему миру. Прокрутите список и выберите ближайшее к вам территориально, и просматривайте папки, пока не найдете самый последний «stage3 tarball» для нужной платформы. (Если вы не уверены насчет своей платформы — i486, i686 или amd64 — пусть за вас выберет результат **uname -m**.) Закончив скачивание, распакуйте архив в свой раздел **root** командой **tar**:

```
tar -xjpf stage3*.tar.bz2
```

Обратите внимание: команда немного отличается от приведенной в справочнике. Я опустил аргумент **v**, который означает «verbose [многословный]» и заставляет **tar** печатать все распаковываемые файлы.

Тут мы подходим к редактированию системных файлов по умолчанию и настройке нашей среды. Первым делом надо настроить компиляторы на оптимизацию и включить/отключить поддержку, например, разных сред рабочего стола. Мы делаем это, редактируя **make.conf**, который живет в **/etc/portage**, не забывая, что мы монтируем внутри **live CD**:



```
* Starting rpcbind ...
* Starting NFS statd ...
* Starting NFS sm-notify ...
* Mounting NFS filesystems ...
* Auto-scrubbing root password for security ...
* Starting autoconfig ...
* Hardware detection started ...
* Detected 1 Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz CPU(s) @ 1664MHz
* Not Loading APM Bios support ...
* ACPI power management functions enabled ...
* No Network device auto detected ...
* Soundcard:
* Intel Corporation 82801AA AC'97 Audio Controller
* driver = snd_intel8x0
* VideoCard: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter
* Adjusting inittab ...
* Starting DHCP Client Daemon ...
* Mounting network filesystems ...
* Starting local

livecd login: root (automatic login)
Last login: Sat Jan 25 15:32:22 UTC 2014 on tty3
Welcome to the Gentoo Linux Minimal Installation CD!
```

```
nano /mnt/gentoo/etc/portage/make.conf
```

В первую очередь в **make.conf** надо отредактировать переменные **CFLAGS**, **MAKEOPTS**. Позднее мы настроим и переменную **USE**. Закончив настройку компилятора, мы готовы ввести нашу подготовленную среду Linux и начать установку. Это называется **chrooting**, т.е. **changing root** [смена root], и означает, что мы по-прежнему будем использовать ядро и поддержку оборудования с **live CD** и делать вид, что наша файловая система **root** — это та, что мы монтировали на **/mnt/gentoo**.

Chrooting состоит из нескольких шагов. Сначала сохраним самое быстрое зеркало Gentoo в **make.conf**, выполнив

```
mirrorselect -i -r -o >> /mnt/gentoo/etc/portage/make.conf
```

и выберем его локально. Далее, проверим, может ли наш прото-**Linux** использовать те же серверы **DNS**, что и **live CD**:

```
cp -L /etc/resolv.conf /mnt/gentoo/etc/
```

Затем быстренько скомануем **ls** и **pci bus**, чтобы сохранить список ваших устройств (через минуту это нам пригодится): **lspci > /mnt/gentoo/lspci.out**.

И, наконец, смонтируем все файловые системы нашей установки, которые заставляют наше оборудование работать, в новый **root**:

```
mount -t proc proc /mnt/gentoo/proc
```

```
mount --rbind /sys /mnt/gentoo/sys
```

```
mount --rbind /dev /mnt/gentoo/dev
```

» **Весь процесс установки целиком проходит из оболочки root, где вы вручную меняете настройку системы командами терминала.**



В Gentoo у вас должен быть отдельный раздел **/boot**, и включать журнал для него незначит, поскольку он изменяется крайне редко; так что лучшая файловая система для использования — **ext2**.

»

Схема разделов

За прошедшие годы отгремело немало дискуссий по поводу схем разбиения диска, и мы здесь вряд ли утихомирим страсти. Однако существует несколько важных моментов, о которых стоит призадуматься, если вы намереваетесь сделать обновление, резервное копирование и изменение/распределение доступа к дистрибутиву более простым.

» **/boot** В Gentoo рекомендуется завести отдельный такой раздел; вы можете форматировать его в **ext2**, чтобы избежать журналирования, и не монтировать этот раздел (или монтировать его только на чтение) для обеспечения безопасности и во избежание всяких происшествий (что при несколько незрелых навыках работы в терминале **root** бывает необходи-

мо). Он не должен быть особенно велик — 128 МБ, вероятно, хватит за глаза.

» **/home** Этот отдельный раздел полезен в двух случаях: если вы загружаете несколько ОС Linux OS и хотите обобщить свой пользовательский профиль; или если вы хотите упростить резервное копирование, и вас беспокоят только данные вашего профиля. Дополнительным бонусом будет, если он находится на отдельном физическом устройстве или на шифрованном или зеркалированном устройстве.

» **Разделы Swap** Никому не нравится паника ядра по поводу отсутствия памяти при открытии 73-й вкладки **Firefox**. Небольшого раздела размером 512 МБ будет вполне достаточно.

Очевидно, что понадобится также раздел **/**; его размер зависит от того, какие приложения вы планируете компилировать и устанавливать, и от того, используете ли вы отдельный **/home**.

И, наконец, не все разделы создаются одинаковыми; из-за ограничений на объем **MBR** на одном диске разрешено создавать только четыре основных раздела (а если на вашем жестком диске уже есть **Windows**, один придется отвести ему). Это не ограничивает количество используемых вами разделов, но вам нужно будет применить «расширенный» раздел, чтобы распространить один или более основных разделов на другие логические устройства. При изменении карты разделов вам нужно точно знать, какой раздел куда идет.

» **Не хотите пропустить номер?** Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Теперь мы готовы к **chroot**:

```
chroot /mnt/gentoo /bin/bash
source /etc/profile
export PS1="(chroot) $PS1"
```

Если в какой-то момент после этого вам надо будет перезапустить процесс установки, надо помнить о необходимости заново совершить некоторые действия, в том числе монтировать ваши разделы, монтировать **proc**, **/sys** и **/dev**, запустить эти команды **chroot** и скопировать **resolv.conf**. Остальные действия, которые сохраняли изменения на диске, нужно совершать только в первый раз.

Прежде чем мы сможем установить какие-то пакеты, надо обновить наш менеджер пакетов до состояния самого свежего репозитория. Менеджер именуется *portage*, и использует команду **emerge**:

```
emerge-webrsync
emerge --sync --quiet
```

(**--quiet** можно опустить, если вы хотите увидеть/сохранить дикое количество результатов.) После обновления можно довершить настройку нашей системы профилем и некоторыми флагами USE. Вы можете просмотреть и выбрать из имеющихся профилей, которые идут с хорошим сочетанием флагов USE:

```
eselect profile list
eselect profile set 2
```

Вам покажут такие вещи, как `desktop/gnome` и `desktop/kde`, и вы должны выбрать тот, который больше соответствует вашим целям (и заменить им **2** выше). Не обращайтесь к профилю разработчика — они не для настройки общей среды разработки, а для разработки самого Gentoo.

Далее, снова откройте **/etc/portage/make.conf** и сделайте свой выбор в переменной USE для более тонкой настройки профиля.

Потом нужно установить часовой пояс и местонахождение. Загляните в файлы часовых поясов в **/usr/share/zoneinfo** и перешлите свой в системный файл через **echo**:

```
echo Europe/Brussels > /etc/timezone
```

А затем сопроводите это кодировкой символов в соответствующем файле настройки: **nano /etc/locale.gen**.

Примеры строк в этом файле будут в `en_GB ISO-8859-1` и `en_US.UTF-8 UTF-8`; проверьте, что включена языковая настройка UTF-8, для предотвращения сбоев. Закончить уместно будет так:

```
locale-gen
eselect locale list
eselect locale set 4
env-update && source /etc/profile
```

И снова, замените **4** нужной вам языковой настройкой.

Час ядра

Как и все остальное в Gentoo, ядро (т.е. часть ПО, которая и называется Linux) нужно компилировать из исходника. Исходники ядра по умолчанию (те, которые разработчики подготовили для наилучшей работы в Gentoo) находятся в *gentoo-sources*; но не переживайте, этот пакет достаточно несложный. Прикажите *portage* найти его: **emerge gentoo-sources**.

В этом пункте справочник дает вам «простую возможность» скомпилировать экземпляр ядра live CD. Предположим, что вы добрались до этой точки и хотите должным образом установить Gentoo, получить молниеносную загрузку и минимальный отпечаток своего ядра в ОЗУ. Итак, вы с нами? Отлично!

Вернемся к команде **lspci**, которую мы запускали раньше, и сделаем заметки о результате: **less /lspci.out**.

Если вам нужен выход вашей материнской платы VGA или работа в вашей новой системе аудио и Ethernet, требуется скомпилировать поддержку ядра для всего этого. Это особенно важно,

поскольку мы также намерены «вычеркнуть» все устройства, которые вам лишние. С другой стороны, большинство устройств работает, если вы включите раздел с правильно указанным производителем; по моему опыту, самая канительная часть — это встроенные беспроводные карты в ноутбуках. Если вы устанавливаете Gentoo на ноутбук и собираетесь использовать Wi-Fi, вероятно, стоит поискать в Интернете результат **lspci** вашей карты для Gentoo, просто чтобы проверить, не принадлежит ли она к проблемным.

Теперь отправимся на поиски исходного кода ядра и откроем основанную на *ncurses* систему настройки:

```
cd /usr/src/linux
make menuconfig
```

Здесь тысячи опций, и когда вы впервые сюда попадете, будет довольно забавно знакомиться с каждой найденной опцией. Оглядевшись здесь, ступайте в раздел Device Drivers и поищите идентификаторы вашего оборудования. Помните, что Chipmaker 8073n Ethernet могут с успехом размещаться в серии Chipmaker 8 в разделе Networking, и тогда вы сумеете отыскать большинство своих самых важных устройств. Если у вас проблемы, не переживайте: вы всегда сможете вернуться и сделать перекомпиляцию, если какое-то устройство будет работать не так, как вы хотите. Еще один важный момент — проверить, включены ли драйверы для всех файловых систем (и что драйверы ненужных файловых систем не съедают место). Удовлетворившись своим выбором, все сохраните, выйдите из меню и скомпилируйте ядро. Эта команда займет некоторое время, так что можете пока что попить чайку!

```
make && make modules_install
```

По завершении можете также скопировать само ядро в раздел **/boot** (перед копированием проверьте версии ядра и архитектуры для вашей системы).

```
cp arch/x86/boot/bzImage /boot/kernel-3.10.25-gentoo
```

И последняя ваша задача, связанная с ядром — найдите все модули ядра, только что скомпилированные:

```
find /lib/modules/<kernel version>/ -type f -iname '*.o' -or -iname '*.ko'
```

и обозначьте их в **/etc**; например, для модулей, перечисленных в первом результате под названием **xt_nat.ko** и **kcopy.ko**, следует изменить **/etc/conf.d/modules**, включив в него **modules="xt_nat kcopy"**.

Если считать установку Gentoo марафоном, то началась финишная прямая; молодцы, что до нее добрались! Прихватите бутылку воды и не сбавляйте темпа. Поскольку мы не хотим трижды вызывать монтирование при каждой перезагрузке, нужно указать системе, какие разделы куда монтировать. Мы делаем это в файле под названием **fstab**, так что **nano /etc/fstab**.

Синтаксис в **fstab** не слишком очевиден, но последовав примеру (отметив ваши разделы **/**, **/boot** и **swap** подходящим устроением **/dev/sd?**), вы вряд ли ошибетесь.

Теперь надо на постоянной основе настроить сетевое соединение на новом компьютере. Дайте своей машине уникальное имя хоста, отредактировав **/etc/conf.d/hostname**, и следуйте рекомендациям справочника, если у вас появятся особо сложные сетевые требования (например, ручная настройка IP вместо DHCP). Далее, нужно выяснить, как, по мнению Gentoo, вызывается наша сетевая карта. Взгляните на выход **ifconfig** и посмотрите, что там указано без названия `lo` (сокращение от "loopback"); возможно, она именуется `eth0`, но к нашей это не относится. Выяснив правильное имя или имена, поставьте их вместо `eth0` здесь:

```
cd /etc/init.d
ln -s net.lo net.eth0
rc-update add net.eth0 default
```

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

ОПЦИИ КОМПИЛЯЦИИ

В файле **make.conf** в **/etc/portage/** вы можете настроить компилятор, который, в свою очередь, настроит все компилируемые вами программы. Gentoo индивидуально настраивает программы при компиляции, а не во время выполнения программы, что значительно ускоряет работу. Вы можете контролировать настройки компилятора, используя некоторые переменные среды в **make.conf**.

CFLAGS (и CXXFLAGS) — дополнительные аргументы, переданные в компиляторы C и C++. Есть разные уровни оптимизации; **-O3** не рекомендуется для системных программ, поскольку он может

вызвать нестабильность, зато **-O2** дает вам широкую оптимизацию без этого недостатка. **-pipe** ускоряет компиляцию, используя больше ОЗУ, передавая промежуточные объекты непосредственно между стадиями компиляции вместо использования временных файлов, а **-march=native** соберет все клавиши быстрого запуска для вашего процессора, которые сможет найти.

В MAKEOPTS, настройка **-jN** скандует **make** запускать процедуры N параллельно, значительно сокращая время компиляции в многоядерных системах. Вы должны настроить N как количество

одновременно запускаемых потоков плюс 1 — это или количество ядер, или, на многопоточном процессоре, удвоенное количество ядер.

Пакеты *portage* слушают переменную USE для изменения поведения в приложениях. Например, если вы указали флаг **-qt** (что означает отсутствие поддержки QT; не путать с флагом **qt**, который как раз включает поддержку), то компилятор может удалить любые относящиеся к QT ветви кода. Профили Gentoo показывают список флагов USE, но вы можете узнать больше из файлов **/usr/portage/profiles/use.desc** и **/usr/portage/profiles/use.local.desc**.

Далее мы должны задать для нашего компьютера пароль root командой **passwd**, настроить раскладку клавиатуры (используя те же ID из последовательности загрузки) в **/etc/conf.d/keymaps**, и убедиться, что нас устраивают настройки в **/etc/conf.d/hwclock** (если у вас двойная загрузка с Windows, проверьте, что указано местное время, а не UTC.)

Теперь пусть **emerge** поработает за нас, делая установку/дополнения к изначальной последовательности всего того, что вам нужно, из системного логгера, демона *Cron* (для планирования повторяющихся процедур), индекса файлов, чтобы заработал поиск по ним, сервера SSH для удаленного доступа и клиента DHCP для получения сетевых адресов от роутеров:

```
emerge syslog-ng
rc-update add syslog-ng default
emerge cronie
rc-update add cronie default
emerge mlocate
rc-update add sshd default
emerge dhcpcd
```

Если вам интересно, можете найти альтернативные пакеты для тех же функций — логгеров и демонов *Cron* полным-полно.

Настройка загрузчика

Конечная (наконец-то) задача процесса установки — установка загрузчика. Вы можете выбрать из нескольких, и лучший выбор (как и с программой разбиения диска на разделы) зависит от того, старый или новый у вас компьютер. *Grub* (именуемый теперь в справочнике «унаследованный *Grub*») или *LILO* отлично подойдут для старых машин, таких, как наш P4, которому нужна MBR (Master Boot Record); но если у вас компьютер EFI, потребуется *Grub2*.

На нашем уроке используется унаследованный *Grub*, но все установки имеют одни и те же этапы. Найдите пакет *portage* —

```
emerge sys-boot/grub:0
и отредактируйте файл настройки:
nano /boot/grub/grub.conf
```

Файлы настройки будут содержать комментарии, объясняющие значение каждой строки. Самая хитрая часть в *Grub* — конвертировать ваши 1-индексированные имена разделов **/dev/sda4** в форму 0-индексированных в *Grub* — (hd0,3) (1 в sda становится 0, а 4 превращается в 3), и — да, эти скобки значащие!

Прежде чем ставить волшебный *Grub* на первые 512 байт системы, убедитесь, что смонтированные разделы указаны в **chroot** — **grep -v rootfs /proc/mounts > /etc/mtab**

```
и тогда загрузчик готов к установке:
grub-install --no-floppy /dev/sda
```

Помните, что если вы производите изменение в файле настройки вашего загрузчика, оно не вступит в силу, пока вы не переписите его в загрузочный сектор — эквивалентом команды **grub-install**.

Ну вот, мы готовы загрузить нашу новую систему Gentoo! Уф! Выйдите и перезагрузитесь.

Если ваша машина загрузилась с первого раза и смогла соединиться с Интернет, примите глубочайшее уважение от команды LXF, о неуязвимый повелитель Эльфов. Мы-то умудрились наделать ошибок на стадии настройки; но главное — не сдаваться! Отладка — это искусство выяснения, какая именно часть чего-то не работает; если вы не видите меню загрузчика, надо всего лишь вернуться в файл его настройки и попробовать другие параметры. Если не работает ваша сеть, возможны проблемы с вашими файлами **/etc/init.d/net.*** — мы просто атомное число раз использовали несуществующий eth0 — или пропущена нужная опция ядра. Но как бы то ни было, вы должны быть в состоянии выяснить, какую часть системы надо проверить дважды, чтобы она не затрудняла процесса загрузки, или чего не хватает после завершения работы.

Чтобы решить большую часть этих проблем, вам нужно следовать стадиям ручного монтирования ваших разделов: **prosc** товарищи, и затем опять **chrooting**, чтобы быть в состоянии все исправить.

Достигнув полного счастья, добавьте учетную запись обычного пользователя, не root:

```
useradd -m -G users,wheel,audio,usb,cdrom -s /bin/bash joe
passwd joe
```

Установка Gentoo — отличная практика: мы заново разбили наш диск на разделы; **chroot**'нули с live CD в систему на стадии полуготовности или сбоя; настроили и скомпилировали собственное ядро; и рассмотрели практически все важнейшие системные файлы Gentoo, многие из которых — общие для всех дистрибутивов Linux. Однако, хоть мы и научились многому, но главная награда — это новый компьютер, который можно будет еще сильнее расширить и индивидуально настроить. Пока что у нас есть только операционная система командной строки, но среды рабочего стола, web-серверы, 3D-графика и прочие ваши любимые приложения Linux готовы к компиляции. Так чего же вы ждете? **LXF**

```
Gentoo Linux -->
[*] 64-bit kernel (NEW)
  General setup -->
[*] Enable loadable module support -->
-* Enable the block layer -->
  Processor type and features -->
  Power management and ACPI options -->
  Bus options (PCI etc.) -->
  Executable file formats / Emulations -->
[*] Networking support -->
  Device Drivers -->
  Firmware Drivers -->
  File systems -->
  Kernel hacking -->
  Security options -->
-* Cryptographic API -->
[*] Virtualization (NEW) -->
  Library routines -->
```

Скорая помощь

Вы можете удостовериться скачивание stage 3 tarball, скачав заодно другой файл, следующий за ним в папке сервера, ***DIGESTS**, и сравнить выходы контрольных сумм по **sha512sum** и **openssl dgst -r-whirlpool** с содержимым этого файла.

» Linux идет с интерфейсом *ncurses* для включения и отключения функций в ядре; самые важные функции — для оборудования и файловых систем.

Calibre: Строим сервер e-книг

Марко Фиоретти показывает, как поделить цифровую библиотеку, чтобы иметь свои электронные книги всегда под рукой, где бы вы ни были.



Наш эксперт

Марко Фиоретти любит свободное ПО. Он учится сам и учит других влиянию открытой цифровой технологии на общество.

Электронные книги — это здорово... если вы храните их в порядке и индексируете. Однако умные менеджеры электронных книг, такие, как *Calibre*, не работают на большинстве портативных устройств для чтения электронных книг (для краткости мы будем далее именовать их читалками). Как же задействовать мощь *Calibre*, чтобы обеспечить присутствие в вашей читалке полной, самой свежей копии всей вашей библиотеки?

Можно постоянно проводить ручную синхронизацию, но это скучно, да и имеет свои ограничения. Однако относительно легко применить серверные способности *Calibre* и других свободных программ для управления цифровой библиотекой и автоматического доступа ко всему ее содержимому и метаданным через Интернет. Вы даже можете поделиться ею с друзьями, родственниками и коллегами — просто убедитесь, что книги и контент, которым вы хотите поделиться, находятся в общественном домене.

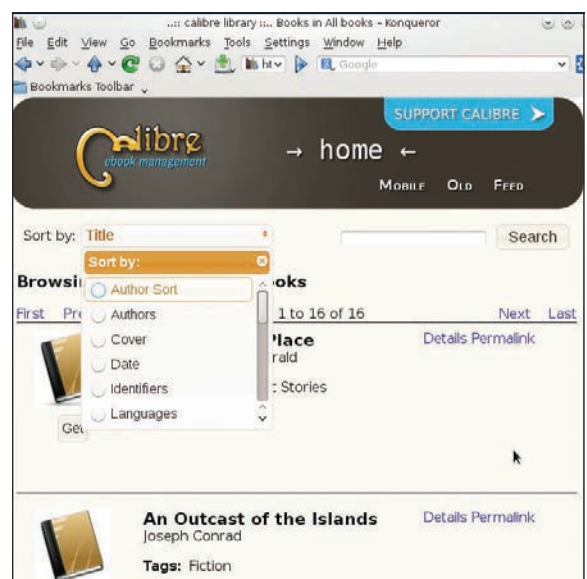
Приступаем к работе с Calibre

Calibre (<http://calibre-ebook.com>) — на сегодняшний день самая полная и известная свободная программа управления электронными читалками. Это многоплатформенное приложение позволяет скачивать книги и соответствующие метаданные из множества онлайн-источников, пометать и сортировать названия и конвертировать их в несколько форматов и напрямую взаимодействовать со многими популярными электронными читалками. Однако все эти задачи *Calibre* выполняет только в локальном масштабе, например, на том же настольном ПК, где установлена программа и хранится внутренняя база данных.

Способ использования *Calibre* по умолчанию для доступа ко всем книгам, хранящимся на вашем ПК, с вашей электронной читалки включает создание прямого соединения (USB или Wi-Fi) между компьютером и читалкой, позволяющего скачивать файлы вручную. Это остается единственным способом сохранить у себя все книги, если вы знаете, что на какое-то время останетесь без доступа в Интернет.

В других ситуациях эта система будет менее эффективна, чем выдача всех ваших книг в доступ через любой браузер или читалку в любое время, без необходимости сначала их скачивать. Например, если та же библиотека *Calibre* должна разделяться между разными читалками одного пользователя (или нескольких пользователей), то имеет смысл создать цифровую библиотеку с централизованным управлением. Это отлично работает для семей, компаний или образовательных учреждений.

Возможно, самое важное, что индексирование, присваивание тэгов и общее управление коллекцией электронных книг с помощью компьютерной программы, независимой от конкретной читалки, экономит время и облегчает создание резервных



» Полная web-, OPDS- или мобильная версия вашей библиотеки *Calibre* может быть рассортирована по нескольким критериям.

копий. Кроме того, это единственный способ обеспечить, чтобы вы не лишились всех своих законно приобретенных книг, если ваша читалка сломается или вы решите приобрести новую модель или сменить бренд.

Стоит рассмотреть возможность размещения менеджера электронных книг — или, по крайней мере, его продуктов — на Raspberry Pi или подобном ему мини-компьютере с Linux, кроме основного ПК. Тогда ваша электронная библиотека будет всегда доступна в локальной сети, даже если ваш основной компьютер будет выключен, окажется вне дома, сломается или будет обновляться.

Домашняя страница *Calibre* прямо рекомендует скачать и установить самую последнюю стабильную версию *Calibre* из командной строки, потому что пакеты, предоставляемые дистрибутивами Linux, «часто ошибочны/устарели». Что бы вы ни думали о таком предупреждении, эта процедура может стать единственным способом заставить *Calibre* работать на сервере.

Для начала проверьте, что на целевом компьютере имеются Python (по крайней мере 2.6 на момент написания) и пакет *xdg-utils*. Если их нет, установите их с помощью стандартных пакетов и процедур вашего дистрибутива.

Затем решите, где вы хотите установить *Calibre*. Обычное местоположение в Linux для программ, устанавливаемых вручную — `/opt` или `/usr/local/bin`, но решение по поводу того, куда

его установить, вы принимаете сами. Программа установки *Calibre* разместит все в подпапке той директории, на которую вы ей укажете.

Сайт также объясняет, как перенести этот параметр в командную строку в случае, если вам нужно будет провести полностью автоматическую установку.

И, наконец, скопируйте, вставьте и исполните в оболочке, от имени `root` или через `sudo`, команду Python, которая показана на сайте. Она спросит вас, куда поместить файлы *Calibre*: скачать их в выбранное вами место и выполнить ряд задач по настройке, например, создать символическую ссылку. Если позднее надо будет удалить программу, просто удалите папку с *Calibre*, созданную в процессе установки. Но помните, что это не избавит вас от символических ссылок.

После установки *Calibre* можно запустить его серверный компонент из графического интерфейса, выбрав в главном меню Content-Share > Start Content Server [Разделение контента > Запустить сервер контента]. Можете также поставить галочку в окошке, чтобы сервер запускался при каждом запуске GUI *Calibre*. По умолчанию *calibre-server*, работающий на ПК, создает простой сайт по адресу <http://localhost:8080>. Здесь вы можете просматривать свою библиотеку по рейтингам, тэгам, авторам или последним данным. Включены подробная информация по каждой книге и ссылки на ее скачивание в имеющихся форматах. Есть также лента OPDS, но о ней мы расскажем на нашем уроке позднее.

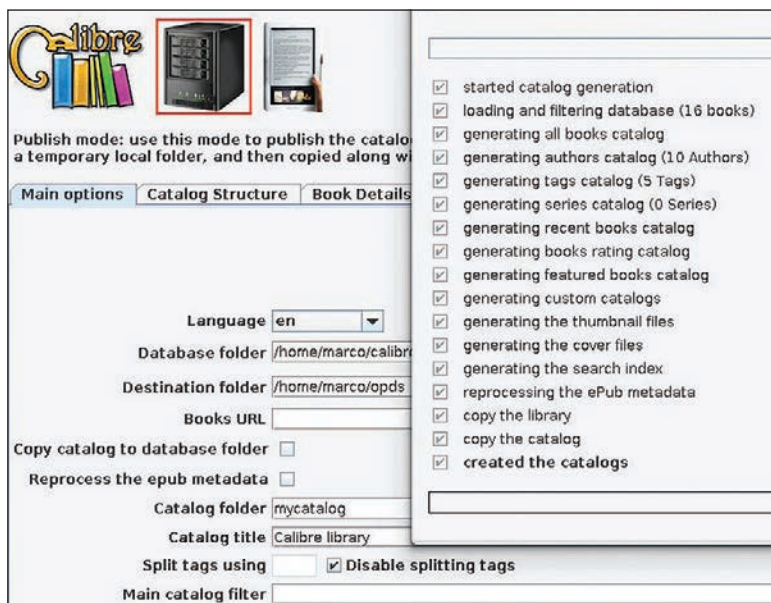
Конечно, работой *Calibre* только локально и только при включенном GUI *Calibre*, проку от нее было бы немного. Серверная функция *Calibre* размещена в программе *calibre-server*. Настройте свой дистрибутив Linux так, чтобы при загрузке он всегда запускал команду

```
calibre-server --with-library=/path/to/calibre-home
```

Здесь **calibre-home** — это папка с базой данных *Calibre* и вашей электронной библиотекой. Web-интерфейс будет доступен в любое время, пока компьютер включен. Программа *calibre-server* принимает опции размера обложки книг и позволяет выбрать имя пользователя и пароль. Есть несколько других настраиваемых параметров, которые четко описаны в <http://manual.calibre-ebook.com/cli/calibre-server.html>. Это все, что вам на самом деле нужно знать, чтобы ваша библиотека *Calibre* была всегда доступна внутри локальной сети.

Поместите свои библиотеки в web

Разобравшись с локальной сетью, давайте посмотрим, как сделать библиотеку *Calibre* доступной вне зависимости от местонахождения ее пользователей. В общем и целом, есть четыре способа добиться этого результата:



➤ При всей своей мощи, *Calibre* способен не на все. Статические web-каталоги, созданные *Calibre2OPDS*, быстрее и имеют намного больше опций.

- 1 Сделать свой локальный сервер *Calibre* доступным через Интернет.
- 2 Установить отдельный *calibre-server* на внешнем (виртуальном) компьютере в хостинг-центре.
- 3 То же, что в пункте 2, но установите *calibre-server* наряду с уже работающим web-сервером.
- 4 Использовать другие программы, чтобы конвертировать базу данных *Calibre* в каталоги HTML или OPDS.

Пожалуйста, не забывайте, что эти классификации довольно условны. Они явно облегчают описание основных различий процедур установки и интерфейсов пользователя, но это на самом деле и вся их цель. Можете свободно использовать эти категории в качестве стартовой точки, а затем смешивать их по своему усмотрению. Например, вполне возможно использовать четвертую опцию на отдельном компьютере. Сделав так, вы получите интерфейс, который:

- » Более гибок, чем предоставляемый *calibre-server*.
- » Работает на web-сервере, которым вы управляете и настраиваете по своему желанию — с паролями, шифрованным соединением и т.д.
- » Всегда доступен для людей вашего круга как из вашей локальной сети, так и из Интернета.
- » Хранит все ваши книги внутри вашего дома или офиса.

Скорая помощь

Каталоги *Calibre*, которые можно просматривать онлайн, будут полезны для семей и организаций. Только убедитесь, что вы делитесь книгами, находящимися в общественном доступе.

Другие «сервероподобные» части Calibre

Наш урок сосредоточен на одном: как автоматически обслуживать электронные книги, управляемые *Calibre*. Сама по себе это очень удобная функция, но основанная на *Calibre* автоматизация умеет куда больше. Несколько кнопок и пунктов меню графического интерфейса *Calibre* — это кнопки быстрого запуска независимых утилит командной строки, перечисленных на странице CLI *Calibre* (<http://manual.calibre-ebook.com/cli/cli-index.html>).

Ничто не мешает вам использовать любую из этих команд отдельно — возможно, в разных скриптах оболочки. Это означает, что как только у вас появится *Calibre*, вы сможете

автоматизировать многие операции, относящиеся к анализу, управлению и генерированию электронных книг. Хорошо, когда вы уверены в том, что делаете; но прежде чем пускаться в эксперименты, не забудьте сделать резервные копии.

Например, программа *fetch-ebook-metadata* делает именно то, что и предполагается по ее названию, и облегчает добавление метаданных книг к любой индивидуальной базе данных, которая вам может понадобиться:

```
#> fetch-ebook-metadata -t 'The Lord of the Rings'
Title : The Lord of the Rings (50th Anniv. Ed.)
Author(s) : J. R. R. Tolkien
```

```
Publisher : Houghton Mifflin Harcourt
Identifiers : amazon:0618640150,
isbn:9780618517657
```

Еще одна область, в которой можно многое сделать, даже не открывая GUI *Calibre*, это конвертирование и форматирование электронной книги. Утилита под названием *ebook-convert* перемещает ваши книги из любого формата, поддерживаемого *Calibre*, в любой другой. *Ebook-polish* «вносит сияние совершенства в вашу тщательно созданную электронную книгу», крайне мало меняя в ее исходном коде: используйте ее, чтобы добавлять обложки и чехлы ко всем вашим книгам.

Лучшая опция для вас будет зависеть от ваших реальных потребностей и от доступных вам компьютеров. Первый метод подходит, если интерфейса web/OPDS, предоставляемого *calibre-server*, достаточно, и вам незачем хранить все свои электронные книги вне своего дома или офиса. Если вы пойдете по этому пути, все, что вам нужно сделать — это настроить интернет-брандмауэр, чтобы разрешить соединения с *calibre-server* (возможно, через нестандартный порт, с помощью опции `--port`).

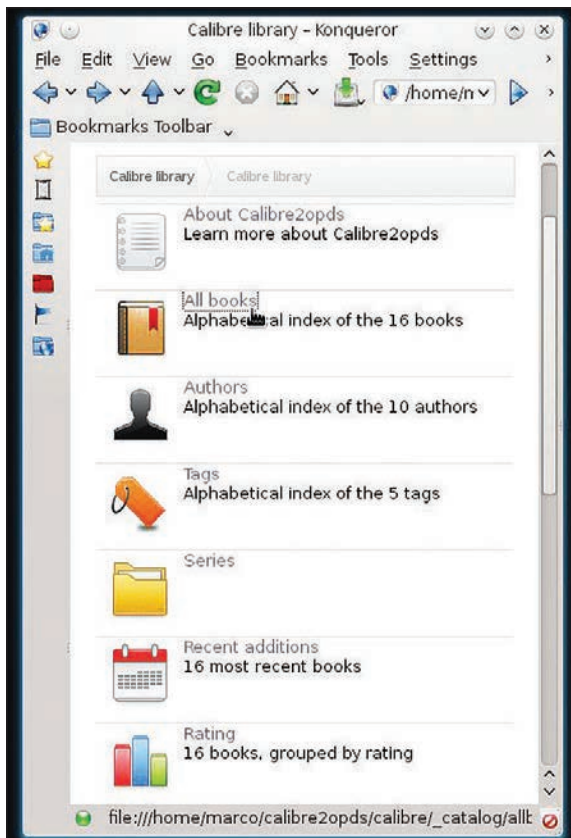
Опции 2 и 3 дают вам такой же интерфейс, но без необходимости каким-то образом открывать вашу локальную сеть. Опция 4 предлагает функции, которых нет в *calibre-server*, и ее главное преимущество в том, что ваша база данных *Calibre* будет совместима с учетными записями недорогого хостинга, которые не позволяют вам устанавливать ничего, кроме статических или основанных на PHP web-страниц.

Давайте рассмотрим опцию 2. Если web-сервер, размещающий вашу библиотеку, имеет доменное имя www.example.com, и вы бы хотели, чтобы ваши книги были доступны на URL www.example.com/mybooks, тогда дайте *calibre-server* команду слушать стандартный порт для web-серверов (80) и используйте префикс `/mybooks`:

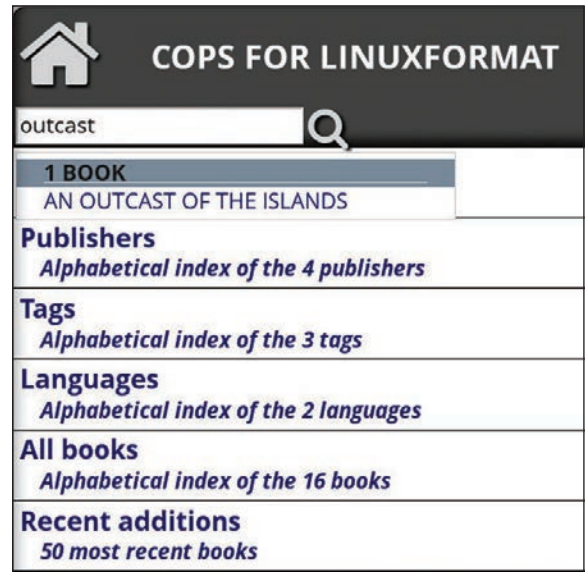
```
calibre-server --port 80 --url-prefix /mybooks --with-library=/path/to/calibre-home
```

У этого решения есть недостатки. Первый в том, что эта команда сработает только в том случае, если нет обычного web-сервера, который уже работает и слушает TCP-порт 80. Вы можете дать *Calibre* команду использовать другой порт, но он может блокироваться корпоративными брандмауэрами. А главное, в незашифрованных соединениях вы сможете только контролировать доступ через базовые функции *calibre-server*: имя пользователя/пароль.

Сценарий номер три предоставляет доступ к *Calibre* из стандартного web-порта, но управляемый (и протоколируемый!)



► Web-версия каталога *Calibre2OPDS* выглядит очень аккуратной и предлагает множество способов сортировки книг.



► Окно поиска COPS очень простое и активировано по умолчанию, в отличие от *calibre-server* или *Calibre2OPDS*.

со всей гибкостью, безопасностью и поддержкой аутентификации настоящего web-сервера. Для этого нужно будет настроить *calibre-server* на нестандартный порт, например, 8000, и затем скомпандовать серверу действовать как обратный прокси-сервер для этого порта. Точные опции настройки для использования с web-сервером *Apache* перечислены со всеми подробностями на <http://manual.calibre-ebook.com/server.html>, и здесь мы только объясним их в общих чертах.

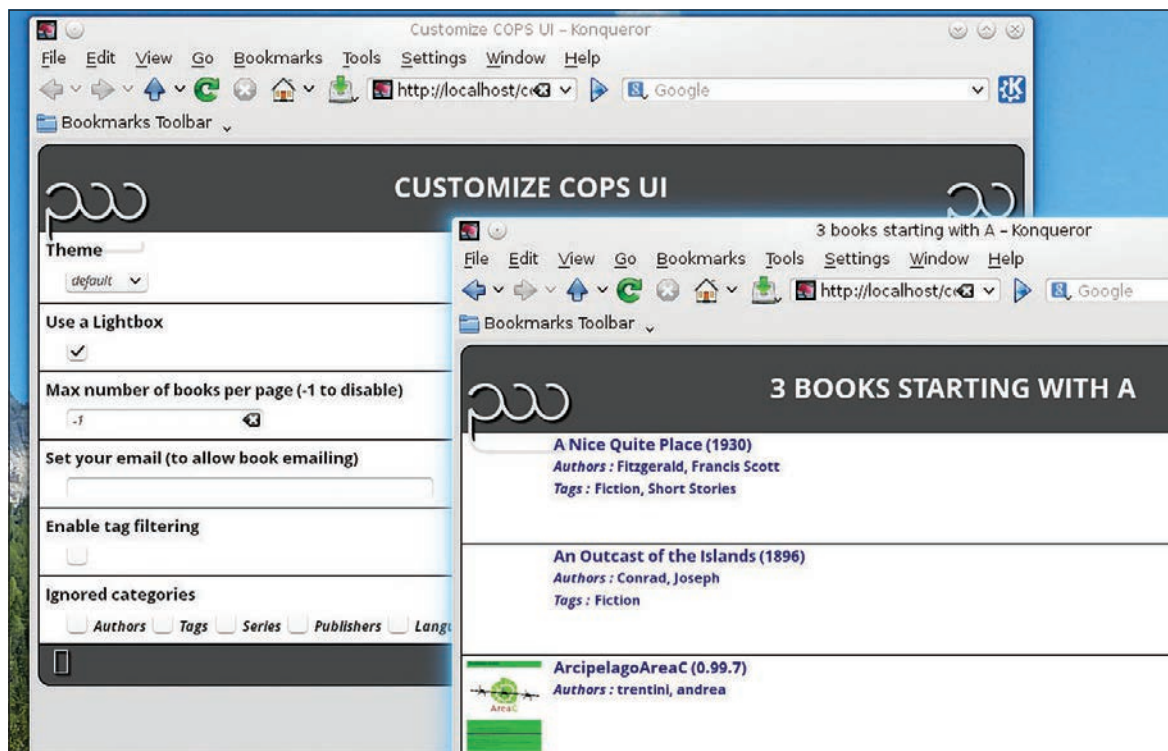
Обратный прокси-сервер просто означает, что когда вы велите своему браузеру загрузить страницу на www.example.com/mybooks, этот запрос получает *Apache* на порт 80, и он передается как есть на *calibre-server* на порте 8000. Какой бы результат ни производился в ответ на порте 8000, он затем передается как есть в ваш браузер. Методы два и три будут работать, только если папка `calibre-home` на вашем сервере является точной копией папки на вашем ПК. Поэтому вам также нужно настроить на вашем ПК задачу *Cron*, которая всегда будет синхронизировать две папки, примерно так:

```
rsync -rvt --delete $HOME/.calibre/ username@servername:/path/to/calibre-home/
```

Серверы Calibre... без Calibre

А теперь перейдем к финальному методу. Без *Calibre*, серверы электронных книг *Calibre* делают содержимое библиотеки *Calibre* доступным онлайн как web-страницы или ленты OPDS. Вы можете получить те же результаты с помощью программ вроде *Calibre2OPDS* (<http://calibre2opds.com/>) или *COPS* (<http://blog.slucas.fr/en/oss/calibre-opds-php-server>), которые генерируют каталоги HTML и/или OPDS с большим количеством функций, нежели в созданных *Calibre*. В отличие от программы-оригинала, *Calibre2OPDS* производит статические web-страницы и ленты OPDS. Поэтому его каталоги будут работать на любой учетной записи хостинга, включая те, что не поддерживают PHP. Однако основное преимущество *Calibre2OPDS* — количество способов, с помощью которых индивидуально настраиваются его каталоги. Во время начальной настройки вы можете настроить — среди прочего — точное количество и комбинацию форматов электронной книги, количество уровней в каждом отображении каталога и размер эскизов.

Можно также дать *Calibre2OPDS* команду добавить к каждой книге ссылки на соответствующие страницы в Wikipedia, Amazon, Goodread, ISFDB и Librarything. Подробное сравнение каталогов,



› COPS считывает базу данных Calibre для создания простых каталогов, настраиваемых в любом браузере.

генерируемых *calibre-server* и *Calibre2OPDS*, имеется онлайн на <http://calibre2opds.com/about/calibre2opds-vs-calibre-catalogs>.

Calibre2OPDS — это приложение Java, которое ни в коем случае не следует запускать напрямую. Пакет для Linux содержит две надстройки оболочки под названием *runGui.sh* и *run.sh*, которые запускают основную программу в режиме GUI или командной строки соответственно.

Графический интерфейс несложен в использовании, пока вы помните о некоторых вещах. Во-первых, щелкните по крупному значку файлового сервера, чтобы перевести программу в режим Publish. Это единственный режим, позволяющий создавать ваш каталог в папке назначения, которая будет содержать все электронные книги и их обложки вне папки *Calibre* (именуемой «папкой базы данных»). Далее, перейдите в Profiles > New profile [Профили > Создать новый] и дайте осмысленное имя профилю использования, который вы намерены создать. Выберите соответствующее место для вашей папки назначения и соответствующее имя и заголовок каталога web-страниц. И, наконец, нажмите на Save [Сохранить], чтобы сохранить все опции в новом профиле, и Generate Catalogues [Создать каталоги].

При первом запуске *Calibre2OPDS* захочет стереть все, что обнаружит в папке назначения. Последующие запуски будут только обновлять то, что там уже находится, но во имя вашего душевного спокойствия — а также более легкого администрирования системы — создайте новую пустую папку специально для этой цели!

На практике мы бы предложили использовать GUI только для изначальной настройки, а затем настроить процедуру *Cron*, которая вызывает *run.sh*, чтобы заново сгенерировать статические каталоги на вашем компьютере, и *rsync*, чтобы скопировать их на ваш сервер.

Гибкие в настройке, каталоги, созданные *Calibre2OPDS*, не имеют функции поиска, поскольку являются статическими страницами. В *calibre-server* есть функция Generate Search Index [Создать поисковый индекс], но она пока что на экспериментальной стадии. Эти проблемы привели к разработке COPS, набора страниц PHP, способных опрашивать базу данных *Calibre*. Результат — аккуратно выглядящий каталог с простым поиском.

Чтобы заставить COPS работать, просто распакуйте Zip-архив с домашней страницы в должную папку вашего web-сервера и сделайте копию файла *config_local.php*. Затем откройте этот файл и измените значение следующих двух переменных на требуемое:

```
$config['calibre_directory'] = './calibrecopy/';
$config['cops_title_default'] = "COPS for LinuxFormat";
```

calibre_directory — путь к полной копии на сервере вашей локальной папки данных *Calibre*. Не забудьте добавить завершающий слэш, иначе COPS не будет работать. Так же, как и с *calibre-server*, вам нужно будет настроить задачу *Cron*, которая будет постоянно синхронизировать эту папку с папкой на вашем ПК. **LXF**

Скорая помощь

Постоянно синхронизировать электронную книгу с *Calibre*? Нет уж, спасибо. Мало того, что это невероятно нудно, так еще и есть ряд серьезных ограничений.

OPDS что?

Если вы используете Интернет больше нескольких месяцев, вы, вероятно, знаете о лентах RSS. (А если нет, то с этим надо что-то делать!) Попросту говоря, RSS — это списки заголовков — со ссылками — самых последних новостей с определенного сайта, в открытом формате, который читается многими программами.

OPDS — сокращение от Open Publication Distribution System (<http://opds-spec.org>), и практически то же самое, что и лента RSS, но специально разработанная для цифровых библиотек. Ленты OPDS предоставляют метаданные многих разных книг в более простом и компактном формате, чем web-страницы, и они даже

удобнее на устройствах, на которых нет web-браузера.

Большинство электронных книг с Wi-Fi или подобными интерфейсами могут автоматически находить эти ленты OPDS в своей локальной сети или брать их из Интернета, чтобы предоставить пользователю полный список книг, которые можно скачать.



Deluge: Ставим торрент-сервер

Маянк Шарма показывает, как *Deluge* превратит старый компьютер, а то и вообще Raspberry Pi в постоянно работающий торрент-сервер.



Наш эксперт

Маянк Шарма любит Raspberry Pi, был редактором Linux.com и писал для LinuxToday и Digg (во времена его популярности).

Если вы похожи на меня, вы всегда думаете, чем бы занять ненужные старые компьютеры. Одно из лучших применений компьютера, который не потянет современную настольную систему — постоянно работающий торрент-сервер без монитора.

С торрент-сервером можно добавлять торренты, управлять и делиться ими с любого компьютера сети. Торрент-сервер можно убрать в угол и оставить свои загрузки ему. Кроме того, при загрузке торрентов постоянно работающий компьютер позволяет поддерживать нормальное соотношение передаваемых и принимаемых данных на вашем трекере.

Для начала подключите к торрент-серверу монитор и клавиатуру (по окончании настройки их можно будет убрать). Пока торрент-сервер подключен к сети через Ethernet или (еще лучше) беспроводное соединение, вы сможете удаленно раздавать торренты и обращаться к загруженным данным.

Сперва нужно очистить жесткий диск торрент-сервера и установить на него свежую копию подходящей серверной ОС, в данном случае Ubuntu Server. Это дистрибутив со скромными системными требованиями, без графического рабочего стола.

Единственное, в чем нужно убедиться — что на компьютере достаточно места для хранения всех загружаемых данных. Для максимальной совместимости с другими системами и устройствами сети лучше всего отформатировать жесткие диски для хранения данных с файловой системой NTFS. Если на этом же диске будет находиться ОС, лучше хранить ее файлы в отдельном разделе.

Торрент-клиентов для Linux имеется несметное количество. В нашем маленьком проекте мы воспользуемся BitTorrent-клиентом *Deluge*, который можно запустить на сервере в качестве демона, принимающего соединения от других клиентов *Deluge*. Клиент *Deluge* кроссплатформенный, благодаря чему к серверу можно будет подключаться как из Linux, так и из Windows и Mac OS X.

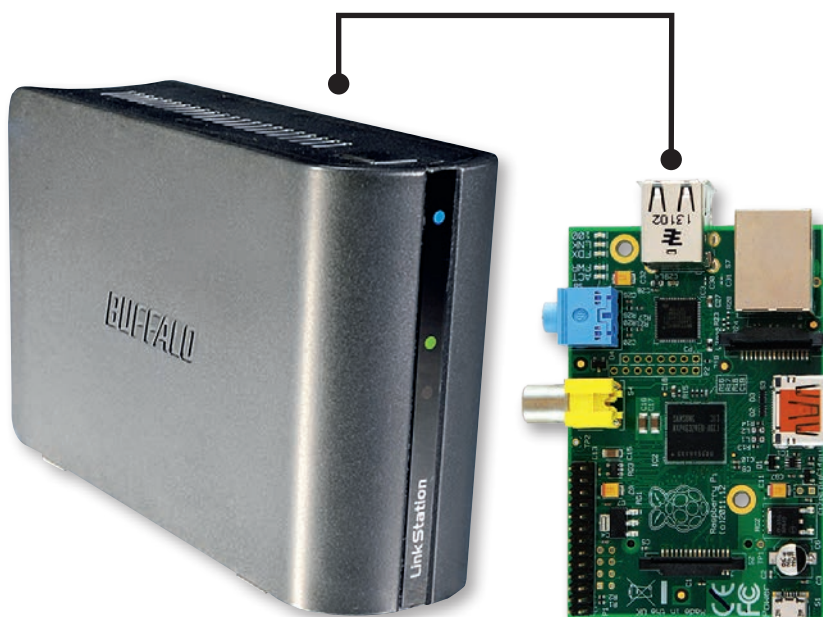
Существует множество способов удаленного доступа к *Deluge*. Базовые функции предоставляются через web-интерфейс. Настроить этот интерфейс нетрудно, но в нем отражена только часть функциональности *Deluge*. Чтобы изучить все функции полностью, нужно настроить демон *Deluge* с других настольных версий *Deluge*.

Торрент-сервер на Raspberry Pi

В качестве торрент-сервера можно использовать и Raspberry Pi. Команды для установки демона *Deluge* и web-интерфейса в этом случае ничем не отличаются от приведенных на уроке. Правда, у Raspberry Pi нет жесткого диска, и придется подключить внешний USB-диск с сервером *Samba*, как описано в пошаговом руководстве (см. стр. 75).

Однако в случае с Raspberry Pi надо выполнить добавочное действие — поколдовать в командной строке, чтобы демон *Deluge* и web-интерфейс автоматически запускались при загрузке Pi. Для этого скопируйте и вставьте два скрипта настройки, предоставленные разработчиками *Deluge* для Ubuntu (<http://bit.ly/1n0Z2w5>). Сделайте каждый из них исполняемым и добавьте в загрузку командами `sudo chmod 755 /etc/default/deluge-daemon` и `sudo update-rc.d deluge-daemon defaults`. Затем включите их в файл `init`: `sudo chmod 755 /etc/init.d/deluge-daemon` и `sudo update-rc.d deluge-daemon defaults`.

➤ Добавим несколько скриптов и превратим Raspberry Pi в выделенный торрент-сервер.



Настройка Deluge

Начнем с настройки web-интерфейса *Deluge*, которая не требует особого труда. Установите необходимые пакеты командой:

```
sudo apt-get install deluged python-mako deluge-web
```

Эта команда автоматически запустит демон *Deluge*, но web-интерфейс придется запустить самостоятельно, набрав **delugeweb &** в терминале.

По умолчанию web-клиент *Deluge* запущен на порту 8112. Откройте браузер и наберите в адресной строке адрес сервера с этим портом, например, 192.168.3.101:8112. Откроется web-интерфейс *Deluge* с полем для ввода пароля. Пароль по умолчанию — **deluge**. После входа в web-интерфейс *Deluge* предложит вам изменить пароль.

Прежде чем начать пользоваться web-клиентом, нужно подключиться к демонам *Deluge*. Web-интерфейс автоматически открывает окно менеджера подключений [Connection Manager], в котором будут перечислены все запущенные демоны *Deluge*. Выберите демона и нажмите кнопку Connect [Подключиться]. Теперь вы сможете добавлять торренты на сервер и управлять ими через web-клиент.

Для изменения порта, на котором работает клиент, нужно изменить его конфигурационный файл. Перейдите в каталог с настройками *Deluge* командой **cd ~/.config/deluge/** и откройте файл **web.conf** в любимом текстовом редакторе. Прокрутите содержимое файла вниз, найдите строку “port: 8112” и замените 8112 на любой порт с номером больше 1000.

Работа демона

Как упоминалось ранее, для использования всех возможностей *Deluge* нужно настроить демон *Deluge* на прием соединений

от удаленного клиента. Начнем с установки консольного клиента *Deluge* для изменения настроек демона *Deluge* — для этого скомаундуйте

```
sudo apt-get install deluge-console
```

Для изменения конфигурации демона остановите демон командой

```
sudo pkill deluged
```

и откройте его файл настройки **~/.config/deluge/auth** в своем любимом текстовом редакторе.

Прокрутите файл вниз и введите три параметра: имя пользователя, пароль и уровень аутентификации демона, разделенные двоеточиями, например, **my-username:my-password:10**. Не забудьте подставить имя пользователя и пароль вместо **my-username** и **my-password**. Уровень аутентификации 10 дает демону административные привилегии.

Сохраните файл и выйдите из текстового редактора. Вернувшись в консоль, запустите демон *Deluge* командой **deluged**. После этого перейдите в консоль *Deluge*, набрав в терминале **deluge-console**. Запустится командная строка демона *Deluge*.

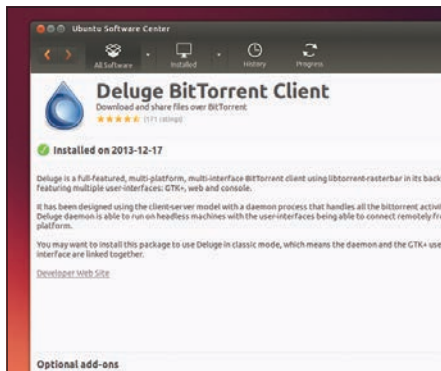
В консоли мы заставим демон *Deluge* принимать удаленные соединения. Для этого наберите

```
config -s allow_remote True
```

Клиент подтвердит, что успешно изменил настройку. Теперь можно выйти из консольного клиента, набрав **exit**, и сделать так, чтобы настройки вступили в силу. Для этого остановите демон командой **sudo pkill deluged** и запустите его снова командой **deluged**.

Вот и все. Теперь *Deluge* будет принимать удаленные соединения от клиентов в сети. Для настройки клиентов воспользуйтесь пошаговым руководством. **LXF**

Подключение к серверу



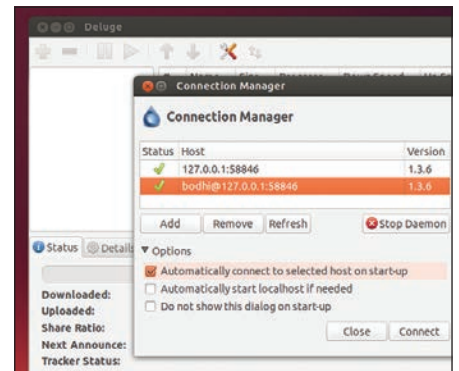
1 Установка клиентов

Теперь демон *Deluge* готов к приему соединений, и нужно установить клиент *Deluge* на остальные компьютеры сети. *Deluge* есть в репозиториях большинства дистрибутивов, так что можете установить его из менеджера пакетов. При необходимости можно зайти на страницу загрузок *Deluge* и скачать клиенты для Windows или Mac OS X.



2 Включаем классический режим

По умолчанию в клиенте *Deluge* настроена загрузка файлов на локальный компьютер. Мы собираемся это изменить, поэтому запустите клиент *Deluge* и зайдите в Edit > Preferences [Правка > Настройки], затем переключитесь в раздел Interfaces [Интерфейсы]. Здесь вы увидите галочку классического режима — Classic Mode, которая по умолчанию поставлена. Снимите ее, нажмите OK и перезапустите клиент *Deluge*.

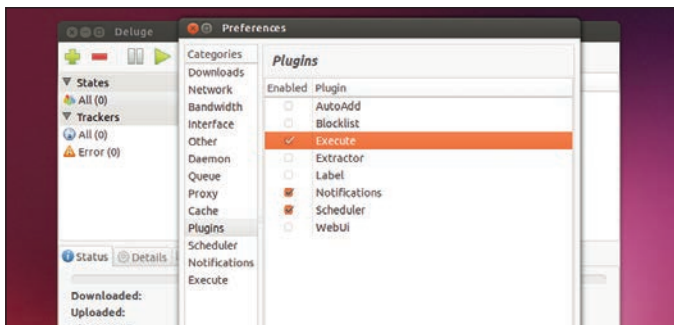


3 Добавляем хост

При перезапуске *Deluge* автоматически запустится менеджер соединений. Нажмите в нем кнопку Add [Добавить] и введите параметры подключения демона. Новое нажатие кнопки Add снова вернет вас в менеджер подключений. Выберите добавленную запись и нажмите кнопку Connect [Подключить] для подключения клиента к демону, запущенному на торрент-сервере.

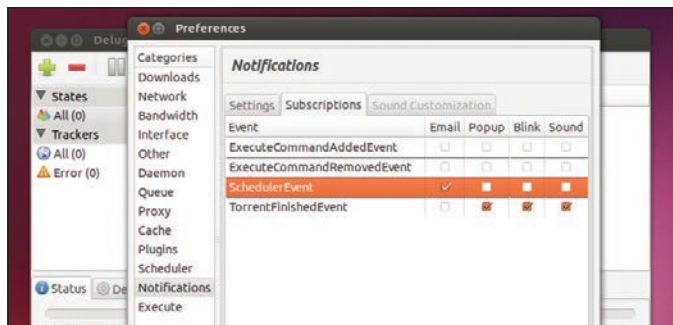
» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Позаботимся об удобстве с торрентами



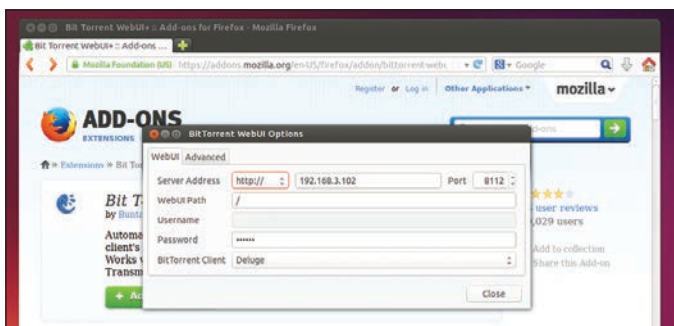
1 Включим плагины Deluge

В *Deluge* немало полезных плагинов. Чтобы включить их, зайдите в Edit > Preferences [Правка > Настройки] и перейдите в раздел Plugins [Плагины]. Для включения плагина поставьте рядом с ним галочку. Включенные плагины появляются как отдельные категории в окне настроек. По умолчанию в клиенте восемь плагинов; для загрузки и установки дополнительных плагинов можно нажать Find More Plugins [Найти дополнительные плагины].



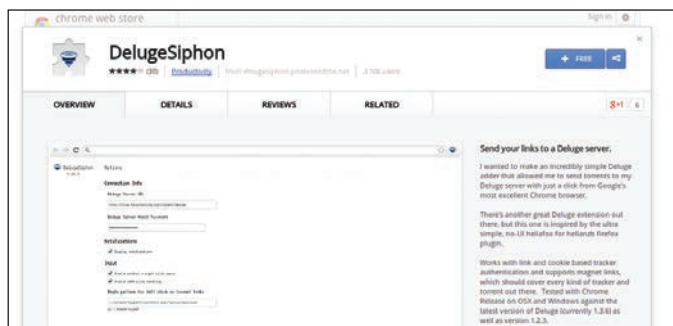
2 Полезные плагины

Один из полезных плагинов — Scheduler: он позволяет ограничивать использование канала в определенное время дня. Плагин Notification оповещает об окончании загрузки торрента, отображая окно на рабочем столе или отправляя вам электронное письмо. Продвинутым пользователям также понравится плагин Execute, запускающий заданную команду при добавлении нового торрента или по окончании загрузки торрента.



3 Расширение для Firefox

Если вы пользуетесь *Firefox*, вам может пригодиться расширение BitTorrent WebUI+ Firefox, оно позволяет добавлять торренты в демон *Deluge* через web-интерфейс. После установки расширения откроется диалоговое окно, где нужно ввести параметры подключения web-интерфейса *Deluge*, и затем при каждом нажатии на ссылку с торрентом тот будет автоматически добавляться на удаленный торрент-сервер.



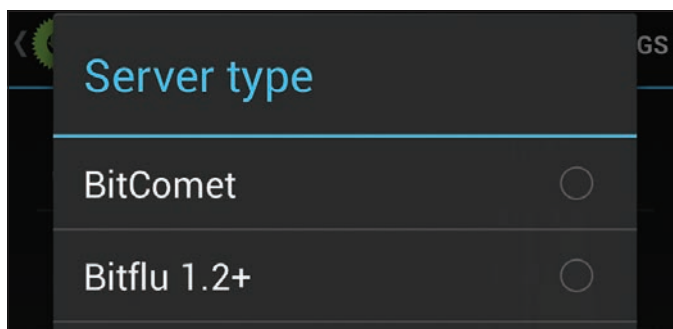
4 Расширение для Google Chrome

Если вы пользуетесь *Chrome*, можете установить приложение *DelugeSiphon* из Google Play Store. Установив расширение, щелкните по его иконке в адресной строке и перейдите на страницу с настройками [Options]. Введите IP-адрес демона, номер порта и пароль web-интерфейса. После этого, щелкнув правой кнопкой на любом торрент-файле, вы сможете выбрать пункт меню Send to Deluge [Отправить в Deluge] для добавления торрента.



5 Используем мобильный клиент

Также можно установить на свой смартфон с Android программу для добавления загрузок и управления ими. В Google Play Store есть несколько приложений, которые могут подключиться к демону *Deluge*. Два самых популярных — *Transdrone* и богатое функциями *TransDroid*. Первое доступно в Play Store, а для загрузки второго нужно зайти со смартфона на его официальный сайт (<http://transdroid.org/latest>).



6 Настройка мобильного клиента

Процедуры настройки *Transdrone* и *TransDroid* одинаковы. При первом запуске программы вы увидите приветственное сообщение. Нажмите кнопку Settings [Настройка] и затем кнопку Add a new server [Добавить новый сервер]. Найдите параметр Server type [Тип сервера] и выберите из списка “Deluge 1.2+”. После этого введите IP-адрес торрент-сервера для подключения к web-интерфейсу *Deluge*.

Используем торрент-сервер TurnKey Linux

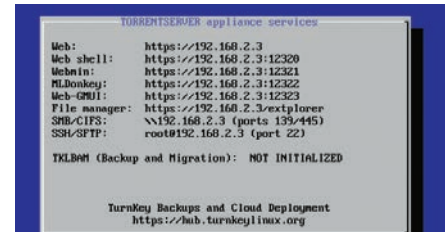
Проект TurnKey Linux занимается созданием автономных специализированных серверов на Linux, работающих на основе компонентов Just Enough Operating System (JeOS), которых, как следует из названия, достаточно для решения конкретной задачи. В образе Torrent Server есть все необходимые утилиты для разворачивания собственного файлового сервера. В нем также есть программа организации доступа к файлам *MLDonkey*, поддерживающая несколько протоколов. С помощью этой программы можно загружать файлы всех типов.

Лучшее в этом образе то, что после установки им можно управлять удаленно через браузер. Можно закачивать и скачивать файлы, упаковывать и распаковывать их с использованием популярных алгоритмов сжатия, типа ZIP и RAR, устанавливать

дополнительные пакеты и делать многое другое из любого браузера в любой операционной системе — как с любого компьютера вашей сети, так и через Интернет.

Образ можно загрузить с www.turnkeylinux.org/torrentserver. Его можно установить сразу или сначала опробовать его компоненты, запустив его в Live-режиме. В любом случае вас попросят ввести пароль пользователя root, а также пароль администратора для различных компонентов образа: программы раздачи файлов *MLDonkey*, *P2P-GUI* для *MLDonkey* и файлового менеджера *eXtplorer*.

Готово. После настройки пользователей образ торрент-сервера скопирует все файлы и установит загрузчик. После перезапуска загрузится консоль



В отличие от нашего сервера с *Deluge*, в торрент-сервере TurnKey применяется более гибкая программа организации доступа к файлам — *MLDonkey*.

конфигурации со списком всех адресов для доступа к различным приложениям сервера.

Хранение загрузок на внешнем диске

```

bodhi@bodhi-Aspire-5738:~$ sudo fdisk -l /dev/sdb
Disk /dev/sdb: 7759 MB, 7759462400 bytes
94 heads, 46 sectors/track, 3504 cylinders, total 15155200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0011b0c8

   Device Boot      Start         End      Blocks    Id System
   /dev/sdb1        2048     15155199     7576576     7  HPFS/NTFS/exFAT
bodhi@bodhi-Aspire-5738:~$ sudo mkdir /media/usb-disk
bodhi@bodhi-Aspire-5738:~$ sudo mount -t auto /dev/sdb1 /media/usb-disk/
mount: only root can do that
bodhi@bodhi-Aspire-5738:~$ sudo mount -t auto /dev/sdb1 /media/usb-disk/
bodhi@bodhi-Aspire-5738:~$

```

```

; preexec = /bin/mount /cdrom
; postexec = /bin/umount /cdrom

[Downloads]
comment = Torrent download folder
path = /media/usb-disk
valid users = @users
force group = users
create mask = 0660
directory mask = 0771
read only = no

```

1 Монтируем внешний диск

Отформатируйте внешний диск с файловой системой NTFS. После этого установите пакеты (`sudo apt-get install ntfs-3g`), чтобы дистрибутив распознавал систему NTFS, и создайте точку монтирования диска командой `sudo mkdir /media/usb-disk`. Подключите USB-диск и найдите соответствующее ему устройство командой `sudo fdisk -l`. Смонтируйте раздел диска (например, `sdb1`) командой `sudo mount -t auto /dev/sdb1 /media/usb-disk`.

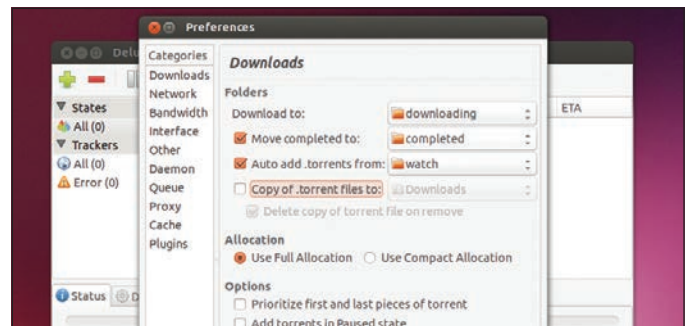
2 Настраиваем удаленный доступ

Для удаленного доступа к USB-диску установите *Samba* командой `sudo apt-get install samba samba-common-bin`. После этого отредактируйте файл настройки — `edit /etc/samba/smb.conf` — раскомментировав строку `security=user`. Создайте в файле новый раздел (идентичный существующему), откройте доступ к разделяемым ресурсам пользователю `downloads` командой `sudo useradd downloads -m -G users` и добавьте его в пользователи *Samba* командой `sudo smbpasswd -a downloads`.

```

bodhi@bodhi-Aspire-5738:~$ sudo mkdir /media/usb-disk/torrents/downloading
bodhi@bodhi-Aspire-5738:~$ sudo mkdir /media/usb-disk/torrents/completed
bodhi@bodhi-Aspire-5738:~$ sudo mkdir /media/usb-disk/torrents/watch
bodhi@bodhi-Aspire-5738:~$

```



3 Создаем каталоги

Затем создайте каталоги, где *Deluge* будет хранить загружаемые файлы. Создайте каталоги для незавершенных и завершенных загрузок командами `sudo mkdir /media/usb-disk/torrents/downloading` и `sudo mkdir /media/usb-disk/torrents/completed`. Создайте еще один каталог командой `sudo mkdir /media/usb-disk/torrents/watch`. Мы настроим *Deluge* так, чтобы этот каталог сканировался и найденные торрент-файлы автоматически добавлялись на сервер.

4 Подключаем их к Deluge

Создав каталоги, следует подключить их к настольным клиентам *Deluge*. Запустите *Deluge* и зайдите в Edit > Preferences [Правка > Настройки]. В окне настроек выберите раздел Downloads [Загрузки], где можно указать каталоги для хранения загрузок. Выберите два первых варианта и укажите каталоги, созданные на внешнем диске.

PHP: Выщарапаем данные с сайта

Кент Ельчук поможет вам запросто извлечь ту самую точную информацию, которая вам нужна, пользуясь вашими личными данными.



Наш эксперт

Кент Ельчук — ведущий web-разработчик в двух колледжах в Ванкувере. В четыре утра, когда у него появляется свободное время, он любит нарывать что-нибудь на сайтах и поиграть со своими Raspberry Pi.

Добытие информации с сайтов [scraping websites] — тонко настраиваемая технология, с помощью которой вы извлечете крупицы золота из любого сайта в Интернете. Причины для охоты за данными множество. Но одна из них — общечеловеческая: это поиск выгодной покупки, и технология позволит вам получить именно то, чего вы хотите. Немного попрограммировав с применением регулярных выражений, вы подберете шаблоны для извлечения необходимых данных с web-страницы — или страниц.

На нашем уроке мы покажем, как прошерстить www.amazon.com насчет стоимости всяких товаров. В примерах мы будем использовать регулярные выражения [Regex]. Регулярные выражения — универсальный метод поиска данных, который помогает найти начала и концы всего, что есть в исходном коде. Освоившись с регулярными выражениями, вы сможете сложить кусочки любой мозаики.

Программировать мы будем на PHP, но так как для анализа используются регулярные выражения, код легко переписать для Perl.

Прежде всего точно определитесь, с какой страницы или страниц вы хотите извлечь данные. Если вы хотите извлечь данные с одной страницы или для одного продукта, шаблоны начала и конца кода будут очень простыми. А вот если нужны сведения о целой категории товаров с широким диапазоном выбора, то для получения информации о нескольких продуктах и страницах, принадлежащих к этой категории, могут потребоваться циклы и различные функции.

Чтобы прогуляться по всему процессу, на первом примере посмотрим, как извлекается информация об одном продукте. Потом двинемся дальше и объясним, как получить информацию об одном и том же продукте у разных продавцов. Основная идея, лежащая в основе извлечения данных с web-страницы —

curl

cURL support	enabled
cURL Information	7.22.0
Age	3
Features	
AsynchDNS	No
Debug	No
GSS-Negotiate	Yes
IDN	Yes
IPv6	Yes
Largefile	Yes
NTLM	Yes
SPNEGO	No
SSL	Yes
SSPI	No
krb4	No
libz	Yes
CharConv	No
Protocols	dict, file, ftp, ftps, gopher, http, https, imap, imaps, ldap, pop3, pop3s, rtmp, rtsp, smtp, smtps, telnet, tftp
Host	x86_64-pc-linux-gnu
SSL Version	OpenSSL/1.0.1
ZLib Version	1.2.3.4

```

root@kent-VirtualBox: ~
└─$ curl
curl
cURL support => enabled
cURL Information => 7.22.0
Age => 3
Features
AsynchDNS => No
Debug => No
GSS-Negotiate => Yes
IDN => Yes
IPv6 => Yes
Largefile => Yes
NTLM => Yes
SPNEGO => No
SSL => Yes
SSPI => No
krb4 => No
libz => Yes
CharConv => No
Protocols => dict, file, ftp, ftps, gopher, http, https, imap, imaps, ldap, pop3, pop3s, rtmp, rtsp, smtp, smtps, telnet, tftp
Host => x86_64-pc-linux-gnu
SSL Version => OpenSSL/1.0.1
└─$
    
```

➤ curl включена, и функциями можно пользоваться.

Получаем удаленный контент и фильтруем вывод

У web-разработчиков и программистов есть разные способы получения желаемого содержимого с web-страниц. Два очень популярных — потоки RSS и API. Первый способ позволяет получить файлы в формате XML, которые можно разобрать и вывести нужные данные. Второй обычно требует

какой-то регистрации для получения логина и пароля, необходимых для доступа к содержимому. Результат часто возвращается в виде массивов, которые, как и любые другие массивы, можно вывести в цикле.

К сожалению, иногда ни один из этих способов не помогает, или вам просто

нужна какая-то ценная информация, которую проще получить самостоятельно. В этом случае поможет поиск и замена шаблона. В PHP функции `preg_match()` и `preg_replace()` позволят найти любой шаблон и заменить любой шаблон необходимыми символами.

Мониторинг ветра перемен

Итак, ура — вы написали программу для получения данных с конкретного сайта, она обшаривает все нужные категории и выбирает искомые данные. Через месяц или два вы вернулись к ней и обнаружили, что никаких данных нет.

Бегло просмотрев сайт, вы увидите, что данные, которые ваша программа должна была накопить, там присутствуют. Раз такое дело, надо бы открыть файл или файлы и определить, в чем загвоздка. Высока вероятность того, что код сайта изменился

и прежние шаблоны перестали ему соответствовать. Пройдясь по коду и шаблонам, вы поймете, как следует изменить программу, чтобы она снова получала необходимые вам данные.

превращение web-страницы в строку. Заполнив строку, вы проходите по ней и собираете требуемые данные. Простейший способ — функция `file_get_contents()`; в других вариантах используются функции `curl`.

Употребим PHP

Функция `file_get_contents()` — простейший способ превратить web-страницу в строку. Освоившись с этой функцией, можно перейти к `curl`, обладающей многими возможностями, которых нет у `file_get_contents()`. Эта функция будет работать, если у вас на сервере установлен PHP.

Для примера возьмем электронную книгу — сгодилась бы любая, но я без ложной скромности раскручу свою: www.amazon.ca/MYSQL-Fundamentals-Snippets-Kent-Elchuk-ebook/dp/B00BSUOD72. У нее есть только одна запись в книгах для Kindle на Amazon. И чтобы узнать стоимость книги, надо разобрать именно эту страницу.

Чтобы узнать, какие данные будут извлекаться, откройте эту страницу в браузере и хорошенько просмотрите исходный код, так как в нем мы будем искать шаблоны и данные для извлечения. В *Firefox* и *Chrome (Chromium)* для просмотра исходного кода достаточно щелкнуть на странице правой кнопкой мыши и выбрать `View Page Source` [Просмотр исходного кода].

Учтите, что исходный код может начинаться со строки в районе 82-й. В исходном коде страницы нужно найти требуемый нам блок кода. Зачем? Ну, глядя на web-страницу, вы видите цену, нечто вроде \$2,84. Ее-то нам и нужно извлечь.

Простое извлечение данных

Поиск по исходному коду показывает, что цена \$2,84 присутствует дважды где-то внутри мешанины символов. Для поиска по шаблону подойдет любое из двух мест, но мы воспользуемся блоком с HTML-тэгом `<input>` и именем `displayedGiftPrice`. Поскольку `displayedGiftPrice` появляется на странице всего один раз, извлечь данные нетрудно.

На простейшем уровне код выглядит так:

```
$data = file_get_contents('http://www.amazon.com/MYSQLFundamentals-Snippets-Kent-Elchuk-ebook/dp/B00BSUOD72/ref=sr_1_1?ie=UTF8&qid=1386253046&sr=8-1&keywords=php+mysql+fundamentals+and+snippets');
/** Получить совпадение по цене */
preg_match('/displayedGiftPrice"[\\s]*value="(.*)"[\\s]*\\>/', $data, $desired_match);
print_r($desired_match);
echo $desired_match[1];
```

Пройдемся по коду. Первая строка, `$data`, содержит URL страницы, данные с которой нужно извлечь. В данном случае это адрес конкретного товара в Amazon.

После этого функция `preg_match` находит нужный блок кода в исходном коде. Она ищет отвечающую шаблону строку и возвращает ее в виде массива. Первая скобка `(.*)` соответствует всем символам между `displayedGiftPrice value="` и завершающей двойной кавычкой `"`.

Фактически эта функция вернет массив с двумя значениями. Первым значением будет вся строка, соответствующая шаблону, а вторым — нужное нам значение; в данном случае, это, разумеется, цена.

В этом примере мы исследуем один продукт и получим его идентификатор, название и стоимость. Вот краткое описание кода, который находит именно те данные, которые нам нужны. Сначала мы получаем содержимое страницы функцией `file_get_contents()`. Затем создается массив `Regex`, из трех элементов — шаблонов для извлечения данных. Первый шаблон определяет идентификатор, второй — имя, а третий — стоимость.

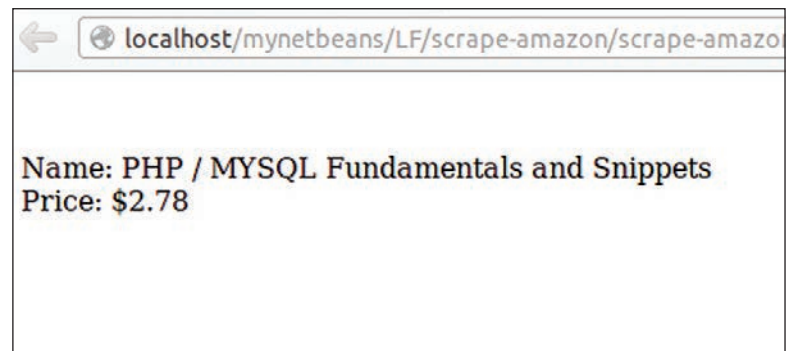
После создания массива каждый элемент массива увеличивается на единицу для получения нужных данных без непрошенных посторонних вкраплений. При каждом проходе цикла в массив `$my_array` добавляется новый элемент. Когда цикл выполнится три раза, `$my_array` будет содержать идентификатор, название и стоимость продукта. Что делать с данными — решать вам. Например, если цена опустилась ниже \$399, можете отправить себе SMS или письмо по электронной почте.

```
$data = file_get_contents('http://www.amazon.com/MYSQLFundamentals-Snippets-Kent-Elchuk-ebook/dp/B00BSUOD72/ref=sr_1_1?ie=UTF8&qid=1386253046&sr=8-1&keywords=php+mysql+fundamentals+and+snippets');
preg_match('/displayedGiftPrice"[\\s]*value="(.*)"[\\s]*\\>/', $data, $desired_match);
print_r($desired_match);
echo $desired_match[1];
foreach ($regexs as $regex) {
    preg_match_all($regex, $data, $posts, PREG_SET_ORDER);
    echo $posts[0][1];
    $my_array[] = $posts[0][1];
}
$regex = '/<span\\s*class="lrg\\s*bold">(.*?)<ul\\s*class="rslt\\s*dkGrey\\s*sim"/';
preg_match_all($regex, $data, $posts, PREG_SET_ORDER);
foreach ($posts as $post) {
```

В этом примере один продукт продается несколькими продавцами. При извлечении данных из нескольких страниц нужно

Скорая помощь

Функции `file_get_contents()` и `curl` умеют загружать web-страницы и превращать их в строки!



» Отображение названия и стоимости в реальном времени.



найти повторяющийся шаблон, пригодный для всех записей в списке.

Общая идея этого упражнения в том, чтобы определить нужные данные и убедиться в том, что их можно отслеживать. И когда цена упадет до нужной отметки, вы получите письмо, SMS и ссылку на продукт и купите его без лишней суеты и переплаты.

Несколько товаров

В этом примере мы покажем, как получить информацию обо всех товарах из раздела электронных читалок Kindle, которые появляются, если набрать “Kindle Fire HDX 8.9” в строке поиска. Этот запрос на www.amazon.com дает семь результатов. Что же с этими семью результатами делать?

Можно просто получить именно то, то мы хотим, а можно создать разные критерии для разных продуктов. Например, на Kindle Fire HDX 8.9" с дисплеем HDX, Wi-Fi и 4G LTE объемом 32 ГБ вы не намерены тратить больше \$529, а на Kindle Fire HDX 7" с дисплеем HDX Display и Wi-Fi объемом 16 ГБ — \$229. Это те товары, которые нас интересуют.

Теперь перейдем к самому интересному — к написанию программы извлечения данных, которая найдет именно эти продукты и проверит их на заданные критерии. Хотя можно было бы сформировать массив адресов и обработать их в цикле, мы уже остановились на том, что будем извлекать данные только с одной страницы. Так как позже на этой странице могут появиться другие товары, с нее стоит начать; к тому же это хорошее упражнение.

URL для поиска “Kindle Fire HDX 8.9” — http://www.amazon.com/s/ref=sr_nr_n_13?rh=%3A2102313011%2Ck3AKindle+Fire+HDX+8.9&keywords=Kindle+Fire+HDX+8.9&ie=UTF8&qid=1386458068&rnid=2941120011.

При анализе этой страницы мы нашли на ней семь товаров. Анализируемый шаблон становится интереснее, потому что у шести товаров цена есть, а у седьмого нет. Эту «мусорную» запись нужно отфильтровать: товар, очевидно, в данный момент недоступен.

Заглянув в исходный код, мы увидим, что у товаров с ценами есть специальный HTML-класс, связанный с товаром, а у товара без цены его нет. Код мог оказаться и таким, что для товара без цены класс существовал бы, только без цены; но тут класс отсутствует. Поэтому универсальный ключ к написанию программы

извлечения данных — доподлинно убедиться, что у каждого элемента есть шаблоны начала и конца, определимые для каждого элемента. В этом случае блоки HTML-кода `` и `<ul class="rsItR dkGrey">` содержат всю необходимую информацию для шаблонов начала и конца.

В отличие от предыдущих примеров с функцией `preg_match()`, здесь мы воспользуемся функцией `preg_match_all()` для поиска нескольких шаблонов на странице.

Вот что происходит. Строку мы получаем с помощью функции `file_get_contents()`. Затем строка `$regex` отображает шаблон, поиск по которому будет выполняться.

```
'/<span\s*class="lrg\s*bold">(.*?)<ul\s*class="rsItR\s*dkGrey/sim'.
```

Шаблон может выглядеть непонятным, но он следует строгим правилам для шаблонов. Во-первых, шаблон существует между двумя прямыми слешами, одним перед `<span` и одним перед `sim`. Вот объяснение всего шаблона.

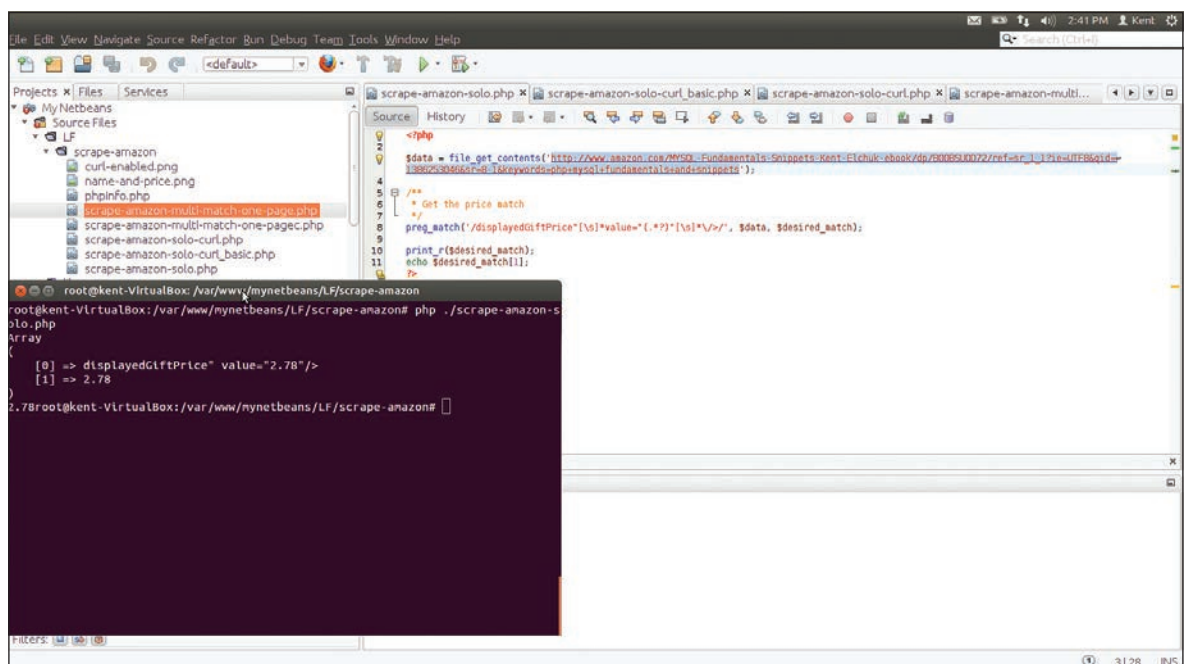
Код начинается со `<span`, за которым следует `\s*`. Это означает «отсутствие любых символов или любое количество пробелов». Так как после `<span` в HTML-коде есть пробел, мы добавляем `\s*`. За этим следует последовательность `class="lrg`, за которой следуют новые пробелы `\s*` и завершающий тэг `bold>`.

Внутри кода

Следующий код в скобках особенно важен: ему соответствует все между предыдущим совпадением и следующим шаблоном, который начинается с `<ul`. Этот фрагмент кода также очень важен, потому что функция `preg_match_all()` создаст массив для каждого соответствия; в одном массиве будет вся строка совпадения, а в другом — код, содержащийся в скобках `(.*?)`. Оставшаяся часть строки соответствия после скобок `(.*?)` включает тэг `<ul`, за которым не следует ничего или произвольное количество пробелов `(\s*)`.

После пробелов определяется заданный класс HTML. Фактически, при поиске данного шаблона HTML-код `class="rsItR dkGrey` становится таким: `class="rsItR\s*dkGrey`. И это завершает наш шаблон.

После второго прямого слэша вы заметите три буквы — `sim`. Это модификаторы шаблона. `S` означает поиск новых строк, `I` — нечувствительность к регистру, `M` — многострочный режим. Поскольку исследуемый блок кода состоит из нескольких строк,



➤ Добавьте `php` в начале файла, и его можно будет запускать из командной строки.

Юридический аспект извлечения данных

Извлечение данных может быть чрезвычайно ценным способом получения и сравнения данных, но его возможностями не следует злоупотреблять. Если вы преследуете чисто личные цели — отслеживать информацию о выгодных покупках и редких товарах и хранить ее у себя — случайные заходы на web-страницу не вызовут подозрений ни у кого. Но если вы затеете, например, извлекать данные со множества крупных сайтов авиаперевозок и публиковать

их в Сети, а особенно — регулярно, вы вполне можете подвергнуться судебному преследованию, как и другие, которые пробовали это делать.

Если коротко, публикация извлеченных данных в Сети и определение самого дешевого товара — «две большие разницы». Многие дела против лиц, извлекавших данные, закончились доказанными обвинениями в нарушении авторского права и материальном ущербе.

Недавно Amazon принял жесткие меры в отношении сторонних провайдеров, использовавших средства извлечения данных с сайта компании без официального разрешения (*Ecommercebytes*, <http://bit.ly/1bjmDCX>).

Так как с помощью программ извлечения данных можно узнать стоимость товара на одном сайте и выставить товар на другом, изменив цену, использование подобных программ — тема весьма спорная.

модификаторы **sim** позволяют снова и снова определять соответствие по шаблону.

С помощью приведенного выше шаблона мы только получаем записи для извлечения данных. Теперь нужно добыть название и стоимость каждого товара и что-то с ними сделать. Для этого массив **\$posts** перебирается в цикле **foreach**, и данные каждой записи тщательно сохраняются для анализа.

Расширение программы

При обработке массива **\$posts** записи обрабатываются последовательно, и каждая имеет имя **\$post**. Затем создается массив **\$regexs** — он найдет название и стоимость продукта для каждой исходной записи. В следующей строке кода создается переменная **\$my_main_block**, это вторая строка в массиве **\$post**.

Так как в исходной программе создается по массиву для каждой записи, в первой строке которого содержится вся строка соответствия, а во второй — только необходимый текст, соответствующий шаблону **(.*)** в первой переменной **\$regex**, мы просто воспользуемся этим блоком кода.

Затем создаются два пустых массива: **\$posts2** и **\$posts3**. Код **\$posts2 = array();** создает пустой массив, так как массивы для каждой записи изначально должны быть пустыми, иначе наш код не сработает.

В другом цикле проверяется проверка названия и стоимости для каждого товара. Цикл выполняется дважды для каждого товара. Мы определяем название товара с помощью следующего кода:

```
'preg_match_all($regex2,$my_main_block,$posts3,
PREG_SET_ORDER);'
```

а стоимость товара — во втором проходе цикла, с помощью кода

```
'preg_match_all($regex2,$my_main_block,$posts2,
PREG_SET_ORDER);'
```

Значения выводятся на экран, если у товара есть и название, и цена. Если чего-то не хватает, значения на экран не выводятся. Код начинается с условия

```
if(!ISSET($posts2) && !empty($posts2) && ISSET($posts3) &&
!empty($posts3))
```

содержащего требования к поиску данных. Обладая названием и стоимостью продукта, можно сделать многое. Две процедуры, которые приходят на ум — сравнить цены с хранящимися в базе данных и отправить SMS; подробнее см. в файле **scrape-amazonmulti-match-one-page.php** на диске.

Теперь мы обзавелись кодом, проверяющим название продукта и цену, и можно добавить в уравнение третий элемент — идентификатор продукта. Когда у продукта имеется уникальный идентификатор, всегда используемый для ссылки на продукт, название и цена уже не так важны, так как первичным ключом всегда будет идентификатор продукта.

В предыдущий пример добавить определение идентификатора легко — по аналогии с названием и стоимостью. После этого можно тайно создать вторую таблицу в базе данных, которая отражала бы изменение стоимости продукта по сравнению со стоимостью продукта при первоначальном добавлении его в список наблюдения.

Например, можно создать таблицу с идентификатором, названием и стоимостью продукта. Затем каждый день будет выполняться задание *Cron* по проверке новых цен. Это сделает обычный запрос *MySQL*:

```
SELECT name FROM original_table WHERE id IN (SELECT id FROM
bargains WHERE price <= '$my_scraped_price' )
```

Если вам захочется развивать этот проект, осталась последняя идея: при появлении сходной цены отправить себе SMS-сообщение или письмо. Многие провайдеры мобильной связи сегодня позволяют отправлять SMS через посредство электронной почты. Письмо часто отправляется на адрес с номером вашего телефона, например, **вашномер@vmobile.ca**. Для отправки такого сообщения можно воспользоваться функцией PHP **mail()** или SMTP. Например, в Perl есть прекрасный класс для отправки почты через SMTP.

Scrape и Cron

В результате нашего урока вы обзавелись базовым набором инструментов для поиска и извлечения данных практически из любого незашифрованного HTML-кода в Интернете.

Извлечение данных с сайтов может быть очень ценным не только для экономии заработанных непосильным трудом денег. С его помощью владелец web-сайта может выполнять регулярное сравнение цен с конкурентами (тем же самым занимаются розничные магазины, только на бумаге) и корректировать свои цены, чтобы они всегда были ниже, чем у конкурентов. Сбор данных также может принести свои плоды при исследованиях. Если есть хорошие сайты с новой, освежаемой информацией, вы можете извлекать эту информацию и получать новые знания.

Наконец, добавив ежедневный запуск такой программы в *Cron*, можно отыскать приработок по своей специальности, или вообще найти новую работу.

```
sudo vi /etc/crontab
```

Например, для запуска задания каждый день в пять часов вечера конфигурация будет такой:

```
0 17 * * * username /usr/bin/php -f /var/www/scrapers/filename.
php
```

Цифры и звездочки (*) в начале строки задают время. Затем указывается имя пользователя **username** и путь к PHP-скриптам на сервере. Наконец, после параметра **-f** указывается файл, который нужно выполнить.

Ручная работа отнимает у вас время. Зачем просматривать web-страницы вручную, если это можно автоматизировать? **LXF**

Скорая помощь

Функция **preg_match()** может найти в строке одно совпадение, а **preg_match_all()** — все повторяющиеся совпадения!

OCaml: Начинаем

Джульетта Кемп показывает, как писать графические программы на OCaml — гибком, высокопроизводительном функциональном языке.



Наш эксперт

Джульетта Кемп находит нечто привлекательное в написании программ на языке, которым пользовался еще ее папа. Хотя и рада, что перфокарты ушли в прошлое.

OCam1 (первоначально “Objective Cam1” — объектно-ориентированный Cam1) — основная реализация языка Cam1, который, в свою очередь, является наследником языка ML родом из 1970-х, предка многих других языков. Как и ML, Cam1 является функциональным языком, и он статически типизирован, что здорово улучшает производительность и позволяет избежать определенного набора ошибок. OCaml добавляет в язык несколько возможностей, в том числе объектную ориентированность, объединяя функциональное, императивное и объектно-ориентированное программирование, что дает очень гибкий конечный результат. Он также повышает быстродействие,

оптимизируя компилятор и предоставляя особенно хорошие реализации определенных функций (например, набор `union`); а благодаря его библиотечной природе программы легко компилировать.

Изучение OCaml позволяет усвоить основные идеи функционального программирования, которые также проявляются в других функциональных языках вроде Haskell, что улучшает общий уровень программирования. Кроме того, функциональные шаблоны очень полезны при решении современных задач программирования, таких как обработка параллелизма. И вообще, OCaml — прекрасная рабочая лошадка: при желании на нем можно даже писать скрипты. Вот и начнем!

Установка OCaml

Теперь у OCaml есть собственный менеджер пакетов, OPAM, который хорошо работает в Linux. С его помощью устанавливать и управлять OCaml и его библиотеками удобнее всего. OCaml также можно установить через менеджер пакетов дистрибутива (*ocaml* — базовый пакет с графикой для Debian и Fedora).

Если у вас хорошая скорость интернет-подключения, установка пройдет довольно быстро:

```
wget http://www.ocamlpro.com/pub/opam_installer.sh
sh ./opam_installer.sh /usr/local/bin
eval `opam config env`
```

При желании можно, разумеется, установить его не в `/usr/local/bin`, а в другой каталог. Для использования `ocamlc` последнюю строку нужно добавить в `.bashrc`, или выполнять эту команду

каждый раз при запуске нового терминала. После установки можно попробовать запустить свою первую программу на OCaml в интерактивной командной строке OCaml. Наберите `ocaml`, чтобы открыть командную строку, и затем следующее:

```
# print_endline "Hello World!";
Hello World
_: unit = ()
# exit 0;
```

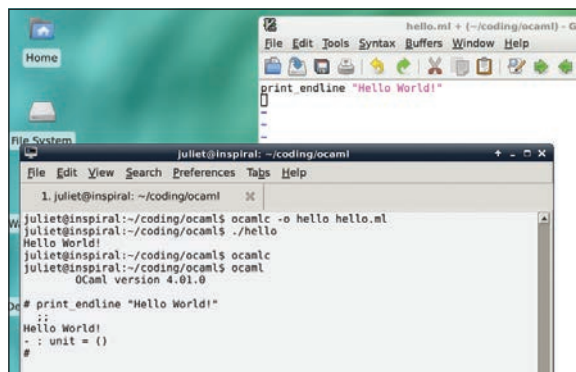
Команды верхнего уровня в OCaml заканчиваются двойной точкой с запятой `;;` (см. врезку *Вопрос точек с запятой* внизу), а аргументы функций OCaml, как обычно бывает в функциональных языках, не ограничиваются скобками. Для выхода из интерпретатора укажите в функции `exit` код возврата в качестве аргумента (в данном случае, ноль, означающий успешное завершение программы).

Наряду с выполнением команды, интерпретатор также сообщает результат и тип результата. В данном случае, результат вызова `print_endline` — это массив (`unit`), единственное значение которого — (). Эта конструкция напоминает функцию `void` в C и используется, когда возвращаемое значение не важно. В данном случае оно не важно потому, что главная задача функции — это вывод строки, а не возврат результата.

Программы можно набирать в интерактивной командной строке, но вы скорее всего будете записывать их в файлы. Создайте файл `hello.ml`:

```
print_endline "Hello World!";
```

и запустите его командой `ocaml hello.ml`. Для компиляции программы воспользуйтесь командой `ocamlc -o hello hello.ml`



➤ Привет, мир! В Vim должен быть по умолчанию установлен файл цветового выделения синтаксиса для OCaml.

Вопрос точек с запятой

В OCaml используются двойная точка с запятой (`;;`) или одна точка с запятой (`;`), а иногда и ни одной.

Вот когда используется каждый из вариантов:

- `;;` используется для разделения команд верхнего уровня в коде.
- `;;` НЕ используется внутри определений функций и внутри команд.
- Иногда `;;` можно пропустить — в частности, перед `let`, `open`, `type` и в конце файла. Во всех этих случаях

OCaml может уверенно определить, что далее следует новое выражение.

Делать это необязательно (и, возможно, лучше уж и не делать — программа будет читабельнее), но такое вам может встретиться в коде, написанном другими.

- `;;` фактически представляет собой оператор **this** и **that** означает «получить два аргумента **this** и **that** и вернуть второй (**that**)». В процессе оба аргумента

будут вычислены. На практике этот оператор используется для разделения команд в блоках кода.

- **let ... in** представляет собой единую команду, и после нее ; указывать не нужно.
- Все остальные команды, кроме последних в блоках кода, должны заканчиваться ;

В кодах нашего урока вы увидите все эти правила в действии. Некоторые операторы `;;` можно было опустить, но для ясности мы их оставили.

Модули

Каждый фрагмент кода OCaml входит в какой-то модуль. Код построения графика, который мы написали в этом модуле в файле `mygraph.ml`, является частью автоматически созданного модуля `Mygraph`. Поэтому в файле можно вызывать любые функции, например, `Mygraph.gtkwindow ()`. Как обсуждалось ранее в этой статье, если нужно вызвать функцию `gtkwindow`, не указывая имя модуля, можно просто

поместить в начало файла строку `open Mygraph`. Это следует делать только если имена функций в модуле не совпадают с именами других используемых функций: в противном случае все равно придется указать полное имя. (Например, имена функций модуля `List` часто используются в других модулях, поэтому обычно никто не указывает в начале модуля `open List`.) Базовая библиотека OCaml очень мала

и содержит лишь простейшие типы и исключения, а также модуль `Pervasives` с операциями над ними. Стандартная библиотека (связываемая с программой автоматически — в программе ее указывать не надо) чуть больше: там массивы, карты, списки и операции вывода. Все прочее надо устанавливать (обычно командой `opam`) и явно привязывать в командной строке, как в коде на нашем уроке.

(после чего запустите программу командой `./hello`). Как правило, код компилируется и запускается таким образом почти всегда. Скомпилированный код работает быстрее; при хранении

программы в файлах ее легко разбить на несколько файлов; а вдобавок при этом проще использовать библиотеки, которые не включены в ядро OCaml.

Рисуем график

Далее на нашем уроке мы напишем программу для рисования графика с помощью нескольких графических библиотек OCaml. В OCaml есть масса очень надежных библиотек, и ими нужно пользоваться при любой возможности. Во-первых, попробуем базовую графическую библиотеку — нарисуем несколько кружков:

```
open Graphics;;
open_graph "640x480";;
for i = 12 downto 1 do
  let radius = 20 in
  set_color blue;
  fill_circle (i * 30) (i * 25) radius
done;;
read_line ();;
```

Для этого нужно открыть библиотеку `Graphics`. Она находится в файле `graphics.cma` — имена пакетов в OCaml пишутся в нижнем регистре, но в обращении их название указывается с большой буквы. `open_graph` — это функция из библиотеки. Здесь она подключается к экрану по умолчанию и создает окно размерами 640×480. При желании можно уточнить экран и расположение окна; более подробная информация указана в документации. Стандартные координаты — математические: начало координат находится в левом нижнем углу окна, `y` увеличивается вверх, а `x` — вправо.

В OCaml есть две версии цикла `for`. В этой, `for... downto...do...done`; значение переменной цикла уменьшается с каждым проходом цикла. Чтобы оно увеличивалось, используйте `to` вместо `downto` и укажите начальное и конечное значения переменной. Тело цикла находится между `do` и `done`;

Очень распространенная идиома в коде на OCaml — **let имя-переменной = выражение in**. Она определяет переменную с текущей областью видимости. Так, здесь переменная `radius` определяется только для оставшейся части блока кода (до следующих `;;`, т.е. для конца цикла `for`). Но по умолчанию значение переменной в OCaml нельзя поменять или присвоить другой переменной: переменные являются неизменяемыми. В функциональном программировании это нормально. В OCaml, как в императивном языке, есть и изменяемые переменные — их мы рассмотрим в следующем разделе, но большую часть времени они здесь нам не потребуются. Неизменяемость переменных позволяет избежать ошибок определенного типа.

Наконец, `set_color` и `fill_circle` — другие функции из библиотеки `Graphics`, которые делают именно то, что следует из названия [Назначить цвет и Закрасить круг]. Попробуйте `draw_circle` [Нарисовать круг] и поглядите, что произойдет. `read_line` ожидает ввода из стандартного потока ввода; эта команда нужна для того, чтобы график оставался на экране. Если ее не будет, программа

закончится, и график сразу же исчезнет. Чтобы скомпилировать этот код, нужно сослаться на графическую библиотеку:

```
ocamlc graphics.cma mygraph.ml -o mygraph
```

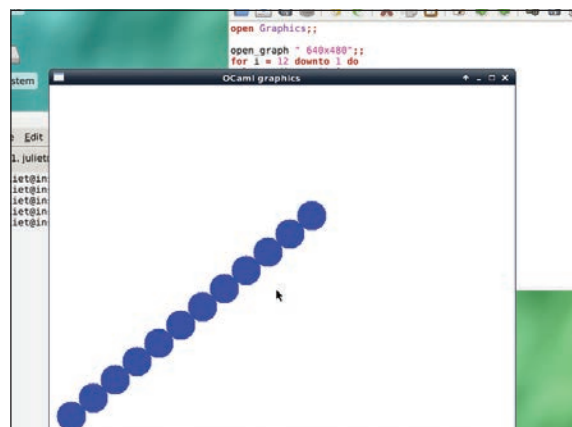
Скомпилируйте и запустите программу, чтобы увидеть набор кружков. Вместо рисования круга с заданной шириной дадим пользователю кое-что ввести и нарисуем график.

```
use Graphics;;
print_endline "Enter value a (integer)";;
let a = int_of_string(read_line ());;
print_endline "Enter value b (integer)";;
let b = int_of_string(read_line ());;
open_graph "640x480";;
for x = 0 to 639 do
  set_color red;
  let y = (a * x) + b in
  plot x y
done;;
read_line ();;
```

Сначала мы спрашиваем у пользователя несколько чисел, используя функцию `read_line ()` по прямому назначению: считывать данные из стандартного потока ввода (с клавиатуры). Потом `read_line ()` возвращает строки, а строку OCaml автоматически не преобразует; надо превратить нашу строку в целое число с помощью функции `int_of_string()` (для преобразования данных между базовыми типами также есть функции `float_of_string()`, `string_of_int()`). Затем мы открываем графическое окно и рисуем точки графика. Цикл `for` перебирает все значения `x` (для окна в 640 пикселей высоты), и для каждого из них задается цвет, определяется `y` как `ax + b` и выводится на экран точка `(x, y)`. Так как `let y` — в цикле `for`,

Скорая помощь

Если хотите попробовать OCaml, не устанавливая его, зайдите на <http://try.ocamlpro.com>, и сможете писать код на OCaml прямо в браузере.



» Кружочки. Попробуйте изменить радиус (изменяя `i`), а также изменить то, что рисуется (например, так: `radius = i * 20 in`).

у существует только внутри этой итерации цикла; **y** можно снова объявить в следующем цикле, и к ней не надо обращаться: ее значение никогда не меняется, каждый раз она объявляется снова.

Следующая строка, `read_line ()`, опять же требуется исключительно для того, чтобы график остался на экране и мы могли его рассмотреть.

Определение функций

OCaml — функциональный язык программирования; но как же определить свою собственную функцию? Перепишем код построения графика со стр. 81 как функцию и одновременно сделаем так, чтобы она принимала на вход вещественные числа (подробнее о которых — через минуту):

```
let a = float_of_string(read_line ());;
let b = float_of_string(read_line ());;
let draw_graph = fun () ->
  for x = 0 to 639 do
    set_color red;
    let y = (a *. float_of_int(x)) +. b in
    plot x (int_of_float(y))
  done;
;;
open_graph "640x480";;
draw_graph ();;
read_line ();;
```

На сей раз мы превратили строку в вещественное число (`float`) вместо целого (`int`). Затем с помощью `let` мы присваиваем метку функции. Конструкция `fun параметры -> [тело функции]`; определяет функцию. В данном случае параметров у функции нет, поэтому мы для ясности указываем `()`.

Тело функции почти не отличается от предыдущего цикла `for`, за исключением того, что в качестве исходных значений и для вычисления используются числа с плавающей точкой; тогда как методу `plot x y` нужны целые числа. `x` создается в цикле `for` как целое число, поэтому при вычислении `y` через `a` и `b` (оба числа вещественные), мы должны преобразовать `x` в вещественное число методом `float_of_int`. При этом сама переменная `x` остается целой, мы просто создаем по ней временную вещественную величину. Так что в следующей строке можно использовать `x` в качестве параметра функции `plot`, а вот вещественную переменную `y` нужно преобразовать в целую величину.

Вы также заметите, что мы используем `.*` и `+.` вместо `*` и `+`. В OCaml арифметические операции (`+` `*` `/`) применяются *только* к целым числам и не перегружаются. Если нужно использовать эти операции с числами других типов (или с объектами любых других типов), вместо них нужно использовать `+. .*` `/.`

Наконец, мы вызываем различные функции. Функцию `draw_graph` можно немного обобщить, заставив ее принимать параметры:

```
let draw_graph = fun a b ->
  for x = 0 to 639 do
    set_color red;
```

```
let y = (a *. float_of_int(x)) +. b in
  plot x (int_of_float(y))
done;
;;
draw_graph a b;;
```

Все, что нам нужно — объявить переменные сразу после `fun`. OCaml сам определит их типы. Возможно, вы заметили, что мы перезаписали глобальные значения `a` и `b` в локальной функции `draw_graph`. Хотя в этом нет ничего страшного, для ясности можно было бы назвать переменные иначе.

Было бы здорово перенести в функцию и ввод данных пользователем, но для этого нам потребуется настоящая переменная: та, значение которой можно изменять и присваивать в любой момент. Пока у нас их не было — мы лишь присваивали переменным имена оператором `let`. `let` сам по себе не создает переменную, он лишь присваивает чему-либо ярлык. Присвоить значение переменной, объявленной с помощью `let`, нельзя — на самом деле это совсем не переменная. Для создания обычной переменной воспользуемся ключевым словом `ref`:

```
let user_a = ref 0.0
let user_b = ref 0.0
let get_user_input = fun () ->
  (* Output user request information to screen before getting
  input *)
  user_a := float_of_string(read_line ());
  user_b := float_of_string(read_line ());
  ;;
  get_user_input ();;
open_graph "640x480";;
draw_graph !user_a !user_b;;
read_line ();;
```

Мы все равно воспользуемся `let` для создания меток (`user_a` и `user_b`). Однако когда мы используем `ref 0.0` вместо присваивания метки переменной, мы присваиваем метку ссылке. Ссылка напоминает указатель в C; теперь метка указывает на то, что хранится в определенной ячейке памяти.

Для присваивания ссылки нужно пользоваться оператором `:=`, а не просто `=`, так как `user_a` — просто адрес ячейки памяти, а нам нужно изменить *содержимое* ячейки. Для получения значения ссылки (ее разрешения) используется конструкция `!user_a` и `!user_b`. Опять же, если вы сталкивались с любым кодом на C или пользовались указателями в других языках, такой синтаксис будет вам знаком.

Также обратите внимание на то, что последняя строка `get_user_input` не заканчивается точкой с запятой, и каждая из двух строк `let` в начале файла не заканчивается `;;`. Точку с запятой можно безопасно опускать в последней строке функции или в конце строки, если следующая начинается с `let`, `open` или `type`. (Подробнее см. во врезке *Вопрос точек с запятой* на стр. 80.)

Использование GTK

Теперь вместо ввода данных пользователем с клавиатуры мы нарисуем графическое окно для обработки этой информации с помощью библиотеки `GTK`. Библиотека `GTK` в OCaml — `lablgtk` — очень неплоха. Установите ее в OPAM командой `opam install lablgtk`. Сначала создадим наше базовое окно в новом файле `mygraph2.ml`:

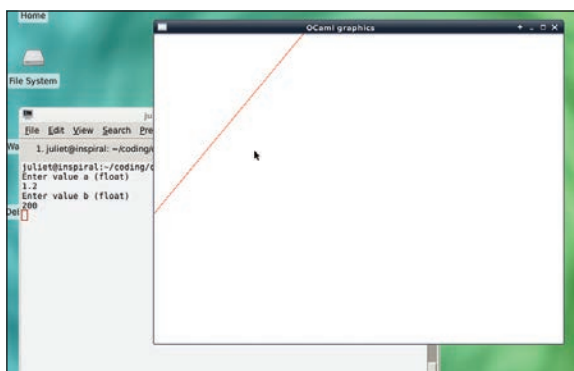
```
open GMain
open GdkKeysyms
```



Скорая помощь

Комментарии в OCaml начинаются с `(*` и заканчиваются `*)`. Они могут быть многострочными.

➤ Указав другие параметры функций модуля `Graphics`, вы можете поэкспериментировать с результатами и нарисовать линии потолще, попиксельно и т.д.



```

let gtkwindow () =
  let window = GWindow.window ~width:320 ~height:240
  ~title:"values" () in
  let vbox = GPack.vbox ~packing:window#add () in
  window#connect#destroy ~callback:Main.quit;
  let entry_a_label = GMisc.label ~packing:vbox#pack () in
  entry_a_label#set_text "Enter value a below";
  let entry_a = GEdit.entry ~max_length:10 ~packing:vbox#pack
  () in
  entry_a#connect#activate ~callback:(get_value_a entry_a);
  entry_a#connect#changed ~callback:(get_value_a entry_a);
  let entry_b_label = GMisc.label ~packing:vbox#pack () in
  entry_b_label#set_text "Enter value b below";
  let entry_b = GEdit.entry ~max_length:10 ~packing:vbox#pack
  () in
  entry_b#connect#activate ~callback:(get_value_b entry_b);
  entry_b#connect#changed ~callback:(get_value_b entry_b);
  let button = GButton.button ~label:"Draw graph!"
  ~packing:vbox#add () in
  button#connect#clicked ~callback:(Main.quit);
  window#show ();
  Main.main ()
;;
gtkwindow ();
open_graph "640x480";
draw_graph !user_a !user_b;
read_line ();

```

Вы видите, что здесь появилось две новых формы обозначений: # и ~.

С помощью # в OCaml реализовано объектно-ориентированное программирование. **entry#connect#activate** аналогично **entry.connect().activate()** в Java. Берется объект **entry**, к нему применяется функция **connect**, и ко всему этому применяется функция **activate**.

Зачем нужна ~? Она позволяет задавать метки для аргументов. Так, в строке

```
let window = GWindow.window ~width:320 ~height:240
~title:"values" ()
```

мы передаем аргументы **width**, **height** и **title** методу **window** в модуле **GWindow**. Эти метки означают, что вам не нужно беспокоиться о порядке аргументов; так мы улучшим читаемость кода. В конце концов, даже не зная об этом, вы могли прекрасно догадаться о том, что делает этот код, правда?

Аналогично в этой строке мы передаем аргумент **callback**:

```
entry_a#connect#activate ~callback:(get_value_a entry_a);
```

Это функция обратного вызова для **activate**.

Мы создаем окно **GTK window** и вертикальное окно **vbox**, которое помещается внутрь первого окна; и мы назначаем окну функцию обратного вызова, чтобы при закрытии окна завершился главный (Main) цикл **GTK**. Главный цикл **GTK** реализует все аспекты жизненного цикла окна **GTK**.

После этого мы создаем метку и задаем ее текст, а также создаем **GEdit** (текстовое поле) для **value a** и **value b**. В окне также есть кнопка **Draw graph** [Нарисовать график]. У полей и у кнопки есть функции обратного вызова, когда их работа закончена, и мы рассмотрим их через минуту. Наконец, мы отображаем окно на экране и передаем управление и главному циклу **GTK**.

Параметр **callback** — это функция, вызываемая при возникновении определенных событий. Так, для текстовых полей мы устанавливаем одну функцию обратного вызова для активации поля (когда пользователь нажимает Enter) и одну для изменения поля (любое нажатие клавиши в поле). Если вместо нажатия Enter пользователь щелкнет мышью вне поля, мы пользуемся **changed**, **activate** в этом случае не сбрасывается. Для кнопки мы просто вызываем **Main.quit** — выход при нажатии кнопки.

Это означает, что код завершит функцию **gtkwindow** и перейдет к следующей вызываемой функции (**open_graph** и **draw_graph**).

Поэтому следующее, что нужно сделать — задать функции обратного вызова.

```

let get_value_a entry () =
  let text = entry#text in
  user_a := float_of_string(text);
  (* prerr_endline text *)
;;

```

В качестве параметра мы передаем **entry** (поле ввода). Напомним, что структура функции обратного вызова такова:

```
~callback:(get_value_a entry_a);
```

В ней определяется функция, которая будет вызвана, и данные, которые будут ей переданы. С помощью **entry#text** мы получаем строку из поля ввода, которую затем преобразуем в вещественное число и присваиваем переменной **user_a**. Закомментированная строка выводит сообщение в стандартный поток ошибок, что удобно при отладке, но не слишком нужно в готовой программе.

Функция **get_value_b** почти идентична предыдущей, но присваивает значение переменной **user_b**. Можно ли как-то обобщить код? Оказывается, можно:

```

let gtkwindow () =
  (* as before *)
  entry_a#connect#activate ~callback:(get_value user_a entry_a);
  entry_a#connect#changed ~callback:(get_value user_a entry_a);
  (* same change for entry_b; rest as before *)
;;
let get_value value entry () =
  let text = entry#text in
  value := float_of_string(text);
;;

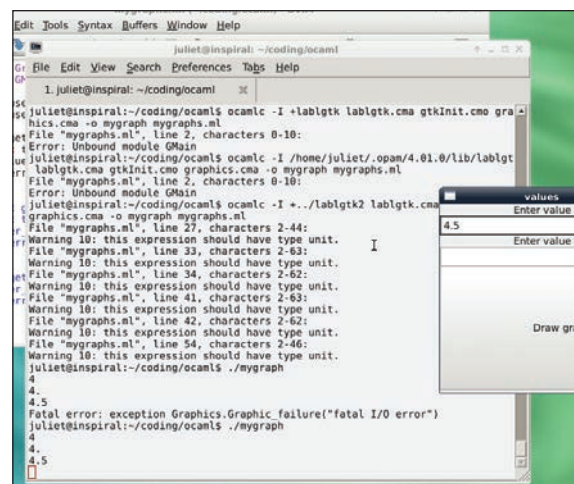
```

Код не требует никаких пояснений. Единственное, что следует отметить — **user_a** надо передавать как ссылку, т.е. без !, решающего ее значение. После этого мы просто присваиваем ей значение в функции **get_value**. Для компиляции этой программы воспользуемся командой:

```
ocamlc -I +. /labgtk2 labgtk.cma graphics.cma gtkinit.cmo -o mygraph mygraph.ml
```

Теперь запустите программу и полюбуйтесь своим новым интерфейсом.

Конечно, здесь мы рассказали только о самых основах языка программирования OCaml, и можно изучить еще массу вещей. Как в функциональном языке, в OCaml есть замкнутые функции (содержащие в себе часть своего окружения), на которые стоит обратить внимание, а также частичные функции. Мы обнаружили, что в документации по OCaml не все отражено полно, но на сайте <http://ocaml.org> есть масса полезных руководств. Удачи! **LXF**



Другой вариант к описанному на уроке — создать структуру, содержащую значения текстовых полей, затем написать метод, который берет структуру и присваивает ее глобальным переменным **user_a** и **user_b**, сделав этот метод функцией обратного вызова кнопки.

У меня были проблемы с правильным вызовом, определяющим положение каталога **labgtk**. В этой версии программы все еще включены отладочные сообщения.

Lua: Начинаем

Lua используется в самых разных областях — от написания игр до обработки фотографий. Прочтите краткое введение в этот язык от **Джульетты Кемп**.



Наш эксперт

Концепция данных как кода в Lua доставляет **Джульетте Кемп** особое эстетическое удовольствие.

Lua — быстрый и легкий интерпретируемый скриптовый язык. Его основная черта — встраиваемость. Движок языка быстр, имеет небольшой размер и легко встраивается в приложения; кроме того, у языка простой API, хорошо интегрируемый в другой код. Язык не особо обременен функциями, но зато имеет метамеханизмы, позволяющие разработчикам реализовать любые требуемые добавочные возможности (но эта тема вне рамок нашего урока). Язык легко освоить, а его возможности легко продемонстрировать в самых разнообразных проектах; и это прекрасный выбор для желающих изучить новый язык.

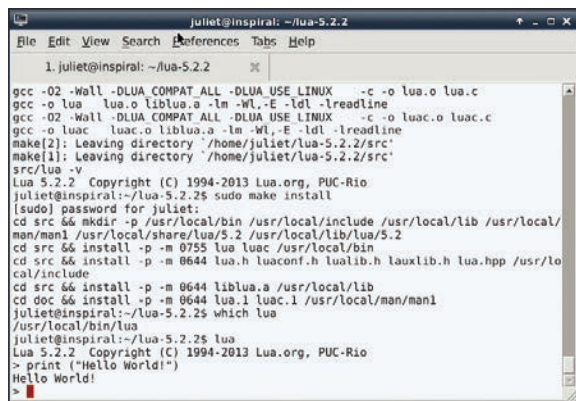
Ола́, Lua!

Для многих дистрибутивов есть двоичные пакеты, но для получения свежей версии Lua (на момент написания статьи 5.2.2) лучше загрузить исходники и скомпилировать их локально. Все просто:

```
curl -R -O http://www.lua.org/ftp/lua-5.2.2.tar.gz
tar xzf lua-5.2.2.tar.gz
cd lua-5.2.2/
make linux test
sudo make install
```

Эти команды соберут пакет и установят его в `/usr/local/bin`. Установив двоичные файлы, откройте терминал и наберите **lua**. Должна открыться интерактивная командная строка Lua. Код можно набирать прямо в ней, например:

```
> print("Hello World")
Hello World
```



```

julieta@insprial: ~/lua-5.2.2
File Edit View Search References Tabs Help

1. julieta@insprial: ~/lua-5.2.2
gcc -O2 -Wall -DLUA_COMPAT_ALL -DLUA_USE_LINUX -c -o lua.o lua.c
gcc -o lua lua.o liblua.a -lm -Wl,-E -lreadline
gcc -O2 -Wall -DLUA_COMPAT_ALL -DLUA_USE_LINUX -c -o luac.o luac.c
gcc -o luac luac.o liblua.a -lm -Wl,-E -lreadline
make[2]: Leaving directory /home/julieta/lua-5.2.2/src
make[1]: Leaving directory /home/julieta/lua-5.2.2/src
src/lua -v
Lua 5.2.2 Copyright (C) 1994-2013 Lua.org, PUC-Rio
julieta@insprial:~/lua-5.2.2$ sudo make install
[sudo] password for julieta:
cd src && mkdir -p /usr/local/bin /usr/local/include /usr/local/lib /usr/local/man/man1 /usr/local/share/lua/5.2 /usr/local/lib/lua/5.2
cd src && install -p -m 0755 lua luac /usr/local/bin
cd src && install -p -m 0644 lua.h luaconf.h lualib.h lauxlib.h lua.hpp /usr/local/include
cd src && install -p -m 0644 liblua.a /usr/local/lib
cd doc && install -p -m 0644 lua.1 luac.1 /usr/local/man/man1
julieta@insprial:~/lua-5.2.2$ which lua
/usr/local/bin/lua
julieta@insprial:~/lua-5.2.2$ lua
Lua 5.2.2 Copyright (C) 1994-2013 Lua.org, PUC-Rio
> print("Hello World!")
Hello World!
>

```

> Вывод команд установки Lua и вывод интерпретатора.

> Такое удобно для первых экспериментов с Lua. А загрузить в интерпретатор код из файла можно командой **dofile filename**.

Написать программу Hello World на Lua очень просто. Создайте файл **hello.lua** со следующим содержимым:

```
print("Hello World")
```

Запустите программу командой **lua hello.lua** — вот вам и привет.

Как видите, синтаксис функций **функция(аргумент)** похож на привычный нам по другим языкам. С помощью **print()** строка легко и быстро выводится на экран. В конец вывода автоматически добавится перевод строки, а аргументы функции будут автоматически преобразованы в строки методом **tostring()**. Однако если вам нужно более тонкое управление выводом, придется воспользоваться методом **io.write()**, который мы рассмотрим позже.

Программа побольше

Программа Hello World была однострочной; перейдем к чему-нибудь посложнее. Напишем простую программу управления временем, которая записывает время начала и окончания задачи. Сначала напишем функцию, регистрирующую время начала выполнения задачи. Создайте файл **time.lua** с таким содержимым:

```

filename = "timefile.txt"
function recordStart(title)
    local timenow = os.time()
    io.output(filename)
    io.write(string.format("%s, start %d", title, timenow)) end
io.stdout.write("Введите имя проекта\n")
local project = io.stdin.read()
io.stdout.write("Введите s для начала отсчета времени или c для отмены\n")
local input = io.stdin.read()
if input == "s" then
    recordStart(project)
else
    error("Старт проекта отменен")
end

```

Первое, на что нужно обратить внимание — порядок команд в файле Lua. Lua — интерпретируемый, а не компилируемый язык, и ему известно лишь о том, что уже встречалось в коде. Это означает, что нужно объявить и описать все функции перед их вызовом. Поэтому в файлах Lua сначала (или в отдельном файле)

Lua: Поставляется в порциях

Каждый исполняемый фрагмент кода в Lua называется порцией [chunk]. Порция — это набор корректных операторов. Это определение относится как к целому файлу с кодом, который будет обрабатываться интерпретатором одна команда за другой, или к одной строке программы в интерактивном режиме.

Функции **main()** или чего-то подобного здесь не имеется — порция будет просто выполняться

как есть. Разумеется, внутри порции можно определять и вызывать функции. Lua может использоваться как язык описания данных (об этом позже), поэтому он легко обрабатывает как очень большие, так и очень маленькие порции.

Полезный прием при отладке — прочесть несколько порций, а потом переключить Lua прямо в интерактивный режим:

```
lua -i -lfile -lotherfile
```

Команда прочтет порции в файлах **file** и **otherfile**, после чего откроется интерактивная командная строка. В интерактивном режиме при каждом нажатии Enter Lua будет незамедлительно вычислять значение введенного выражения. Если введенное выражение логически не завершает порцию, Lua будет ждать следующего. Это значит, что в интерактивной командной строке можно вводить многострочные определения — например, для функций.

объявляются функции, а за ними следует основная логика программы. **filename** — глобальная переменная. В Lua все переменные глобальны, если явно не определены как локальные (с помощью выражения **local**, как в переменных **local input** и **local project**). Если переменная используется только в блоке, где определена (и никогда — в прочей части кода), ее стоит объявить локальной. Так будет аккуратнее, быстрее, и снизится вероятность ошибок.

В нижней части основного кода вы видите функцию **io.write()**, которая, как мы уже говорили, является аналогом **print()**. (Но придется добавлять переход на новую строку в конце строки вывода.) По умолчанию **io.write()** будет использовать «текущий» файловый дескриптор, которым будет **stdin/stdout**, если вы не меняли его. Но чтобы гарантированно вывести данные именно туда, куда вы хотели, можно указать это явно: **io.stdout.write()**. При вводе данных из стандартного потока ввода **io.stdin.read()** данные считываются до первого перехода на новую строку. Вы также заметили управляющие операторы **if / then / else / end**. В Lua условие для **if** указывается без скобок, а **then** должен быть в той же строке.

Наконец, в **recordStart()** вы видите строку **io.output()** — она перенаправляет вывод по умолчанию в заданный файл. После этого в файл можно записать данные, отформатировав их с помощью **string.format()**. Функция **os.time()** возвращает количество секунд, прошедших с начала эры Unix [с полуночи (00:00:00 UTC) 1 января 1970 года, — *прим. пер.*], что прекрасно подходит для наших целей. Если потребуется его вывести, его можно будет преобразовать с помощью другой функции из библиотеки времен.

Несколько раз запустив эту программу командой **lua time.lua**, вы увидите, что файл каждый раз перезаписывается. Поэтому начать новую задачу, не завершив старую, вы не сможете. Чтобы данные добавлялись в существующий файл, укажите это явно:

```
function recordStart(title)
    local timenow = os.time()
    local f = assert(io.open(filename, "a"))
    f.write(string.format("%s, start %d\n", title, timenow))
    f.close()
end
```

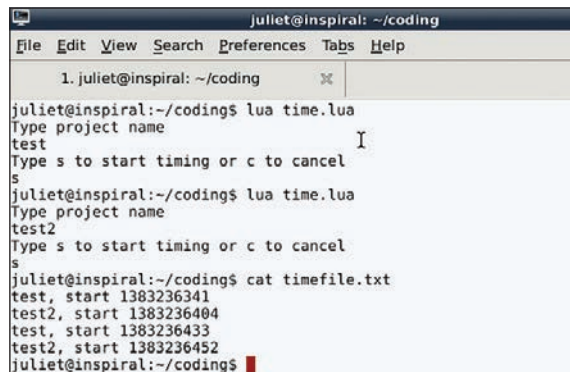
Строка **assert(io.open())** — популярная идиома Lua. **io.open()**, как несложно догадаться, открывает указанный файл в заданном режиме (параметр **a** здесь означает «добавление [append]»). Результат вызова этой функции записан в переменную **f**, в которую затем можно записывать данные как прежде. Однако обертка ее в **assert()** означает, что если вызов **io.open()** завершится неудачно, ошибка будет передана в метод **assert**, который выведет соответствующее сообщение.

Теперь вы сможете начинать несколько задач. Проверьте, что они появляются в выходном файле... и перейдите к следующему разделу.

Завершение работы

Итак, мы можем начать сколько угодно проектов, но они у нас не кончаются. Поэтому наша следующая задача — добавить функцию для регистрации времени окончания:

```
function recordEnd(title)
    local timenow = os.time()
    local f = assert(io.open(filename, "a"))
    f.write(string.format("%s, end %d\n", title, timenow))
    f.close()
end
-- Запрос на ввод как сделано выше
if input == "s" then
    recordStart(project)
elseif input == "t" then
    recordEnd(project)
else
    error("старт проекта отменен")
end
```



end
В основном теле программы мы добавили в оператор **if** еще один вариант — **elseif input == "t"**. В блоке **if** может быть сколько хотите вариантов **elseif**. Остальная часть кода нам хорошо знакома.

Однако данные не добавляются в существующую строку, а просто добавляется новая строка. А нам нужна вся информация в одной строке. Чтобы этого добиться, нужно прочитать строку обратно и затем изменить ее. Еще раз посмотрим на функцию **recordEnd()**. Для удобства разобьем ее на более мелкие функции:

```
function editLine(line)
    local timenow = os.time()
    return line .. string.format(", end %d", timenow) end
function writeWholeTimeFile(lines, linesRemain)
    local f = assert(io.open(filename, "w"))
    for i, line in ipairs(lines) do
        f.write(line, "\n")
    end
    if linesRemain then f.write(linesRemain) end
    f.close() end
function recordEnd(title)
    local f = assert(io.open(filename, "r"))
    local lines = {}
    local linesRemain
    for line in f.lines() do
        if string.find(line, string.format("^%s", title)) then
            lines[#lines + 1] = editLine(line)
            linesRemain = f.read("*a")
            break
        else
            lines[#lines + 1] = line
        end
    end
    f.close()
    writeWholeTimeFile(lines, linesRemain)
end
```

К сожалению, изменить одну строку в Lua непросто. Вместо этого мы читаем все, затем снова выводим все, перезаписывая файл с одной измененной строкой.


В функции **recordEnd()** мы открываем файл и устанавливаем несколько переменных. **lines** — это массив. У массивов в Lua нет фиксированного размера, поэтому все, что вам нужно для инициализации массива — объявить его, как здесь. Нумерация элементов может начинаться с любого числа, но по умолчанию (в том числе в стандартных библиотеках) она начинается с 1.

Цикл **for** проходит по файлу с помощью библиотечной функции **lines()**. Для каждой строки мы проверяем, не начинается ли она с названия интересующего нас проекта (подставляя его в поисковую строку с помощью **string.format()**). Если нет (блок **else**), мы добавляем ее в наш массив **lines** и переходим к следующей строке. **#lines** возвращает количество строк в массиве, и если индексация начинается с 1, **#lines + 1** — следующий пустой элемент массива. »

» Запустите код несколько раз и проверьте выходной файл, дабы убедиться, что все в порядке.

Скорая помощь 

Комментарии в Lua начинаются с **--** и могут вставляться прямо посреди строки.

Скорая помощь 

Конструкция **foo.method()** означает запуск метода **method()** объекта **foo**. В то же время **io** — библиотека, а **io.stdin** — конкретный объект (фактически — файловый дескриптор), соответствующий стандартному потоку ввода системы.

Скорая помощь

Для работы с фактическим временем можно использовать функцию `os.date()` с различными параметрами формата для преобразования количества секунд с начала эры Unix в человеко-читаемый формат времени. Однако с продолжительностью проекта это не работает, так что выполните преобразование самостоятельно.

Если текущая строка начинается с обязательного названия проекта, мы изменяем строку с помощью `editLine()` (используя оператор конкатенации `..`), затем помещаем измененную строку на следующее место в массиве. Затем мы считываем все остальное содержимое файла (`*a` означает `*all`) в переменную `linesRemain` и выходим из цикла. Закрыв файл, мы вызываем функцию `writeWholeTimeFile()`, чтобы снова записать весь файл.

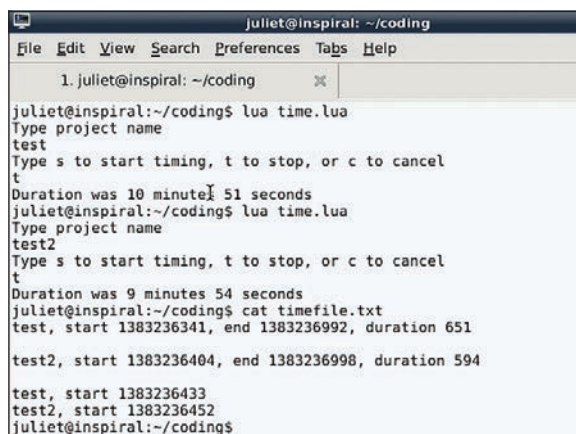
В функции `writeWholeTimeFile()` используется режим открытия файла `w`, а не `a`, потому что на сей раз мы целенаправленно хотим перезаписать файл. Единственное изменение в коде здесь — `ipairs()`. Это компактный элемент синтаксиса Lua, который проходит по всем элементам массива. `ipairs` возвращает два значения: индекс в массиве (здесь — `i`, мы его не используем) и значение из массива (здесь — `line`). Записав весь массив, мы выводим все остальное из `linesRemain`, если там вообще есть значения.

Мы улучшили предыдущую версию программы, но все равно не знаем, сколько времени ушло на решение задачи! Давайте определим это время, выведем его на экран и запишем в файл. Для этого потребуется изменить лишь функцию `editLine()`:

```
function editLine(line)
    local timenow = os.time()
    local _, _, starttime = string.find(line, "%w+", start (%d+)%)
    local duration = timenow - starttime
    print(string.format("Duration was %d minutes %d seconds",
        duration/60, duration % 60))
    return line .. string.format(" end %d, duration %d", timenow,
        duration)
end
```

Переменная `_` — это переменная-пустышка. Нам не интересны первые два значения, возвращаемые функцией `string.find()` — индекс начала соответствия и индекс окончания соответствия. На самом деле, обычно `string.find()` возвращает только два значения, так откуда же взялось значение `starttime`? (Соответствия обрезаются во врезке «Шаблоны поиска».)

Определив время решения задачи, мы выводим его, чтобы показать пользователю, долго ли он работал (% — оператор деления по модулю, им удобно пользоваться в функциях работы со временем).



➤ Остановка таймера. Лишние пустые строки в выходном файле — результат ненужных новых строк (они не включены в приведенный здесь код).

Мы теперь умеем запустить таймер, остановить его и узнать, как долго мы работали над проектом. Однако сейчас мы можем записать длительность только одной задачи для одного названия проекта. Было бы здорово иметь возможность сохранять несколько задач для одного проекта. Кроме того, было бы хорошо записывать информацию более удобным способом.

Таблица времен

В нашем случае можно было бы хранить данные в CSV-файле и затем разбирать его, но это малость уродливое решение. Лучшая (и более свойственная Lua) альтернатива — воспользоваться таблицей. На самом деле, таблицы — это единственная структура данных в Lua (даже массивы по сути представляют собой таблицы). Таблицы очень гибкие, их легко создавать и ими удобно пользоваться. Мы создадим таблицу, в которой ключом будет номер, а значением — другая таблица, содержащая данные проекта:

```
{
    1 -> {project = NAME, starttime = TIME, endtime = TIME, duration = DURATION}
    2 -> {project = NAME, starttime = TIME, endtime = TIME, duration = DURATION}
}
```

За счет механизмов обработки массивов в Lua это массив таблиц с уникальным числовым ключом у каждой таблицы. Это означает, что для одного проекта можно создать несколько записей. Как мы уже отметили, наша «таблица» данных на самом деле представляет собой массив таблиц, поэтому метод `readInTable()` работает именно так, как должен.

Сначала напишем новую версию кода, который записывает время начала проекта, со вспомогательной функцией для вывода таблицы (это не обязательно, но удобно для отладки):

```
timetable = {}
function tableRecordStart(title)
    local timenow = os.time()
    table.insert(timetable, {project = title, starttime = timenow})
end
function outputTableToScreen()
    print "Printing table"
    for i, v in ipairs(timetable) do
        print(i, v.project, v.starttime, v.endtime, v.duration)
    end
end
-- Главный код как раньше, с такой заменой:
if input == "s" then
    tableRecordStart(project)
elseif ...
-- Остальное как раньше
outputTableToScreen()
```

В начале файла создается глобальная переменная `timetable`. Функция `tableRecordStart()` очень проста: в ней мы просто пользуемся функцией `table.insert()`. Если вызывать ее с двумя аргументами как здесь, то новая строка добавится в конец таблицы. (Или можно указать, куда именно ее вставить.) Обратите внимание, что пока мы не создаем переменные `endtime` и `duration`. В Lua таблицы

Шаблоны поиска

Вот вызов функции `string.find()` в коде:

```
local _, _, starttime = string.find(line, "%w+", start (%d+)%)
```

Мы ищем всю строку, указанную между кавычками, т. е. `%w+`, `start (%d+)%`. Здесь используются регулярные выражения Lua. Это не то же самое,

что регулярные выражения для POSIX, хотя у них много общего. Как и в POSIX, `%w+` означает «одна или несколько буквенно-цифровых символов», `%d+` — «одна или несколько цифр», а `%$` означает конец строки. Скобки означают границы шаблона, и именно отсюда мы получаем `starttime`. Если

найдена полная строка соответствия, то все, что между скобками, записывается в `starttime`. Это должна быть строка цифр — записанное время начала проекта в каждой отдельной строке.

Тот же синтаксис используется в `string.gsub()` и в других функциях поиска строк по шаблону.

можно расширять при необходимости, и эти значения можно добавить, когда они у нас появятся.

В функции вывода для итерации по таблице снова используется `ipairs`. `i` — числовой ключ, а `v` — значение, которое в данном случае представляет собой другую таблицу. Вы видите, что точно так же происходит итерация массивов, подтверждая, что массивы по сути являются таблицами.

Существует несколько способов получения информации из таблицы — один из них таков: `table.name`. Эта синтаксическая обертка для `table["name"]`, что означает таблицу, индексируемую строкой `"name"`. Если бы мы написали `table[name]`, то получили бы таблицу, индексируемую переменной `name`. Вы должны понимать, что это не всегда одно и то же:

```
name = "foo" table[name] = "bar" -- это таблица table["foo"]
name = "foo" -- это table["name"]
```

При создании строки таблицы с помощью синтаксиса `project = title`, как в функции `tableRecordStart()`, `project` считается строкой. Общая мораль такова: внимательно следите за своими ссылками и индексами!

Запустив этот код, мы сможем ввести время начала проекта, и в конце оно будет выведено, но потом таблица исчезнет в конце программы. Это означает, что нельзя добавить ни одного времени окончания проекта, так как нет ни одного времени начала проекта, с которыми они будут связаны.

Следующая задача — сериализовать таблицу. Здесь «порции» Lua особенно удобны. Если мы сериализуем таблицу так:

```
readInTable{project = "test", starttime = "1383216178",endtime = "1383216665", duration = "487"}
readInTable{project = "test", starttime = "1383216714",endtime = "1383216731", duration = "17"}
```

а потом просто запустим этот файл как код Lua, будет запущена функция `readInTable()` с указанной таблицей в качестве аргумента. Поэтому можно создать нашу таблицу так:

```
function readInTable(o)
    table.insert(timetable, o)
end
dofile(filename)
```

`dofile(filename)` считывает указанный файл и выполняет его как код Lua. Строки файла считываются одна за другой и отправляются методу `readInTable()` (`o` — таблица, передаваемая как аргумент), где вставляются в конец таблицы `timetable`. Очень элегантное решение. Нужно лишь правильно сериализовать таблицу:

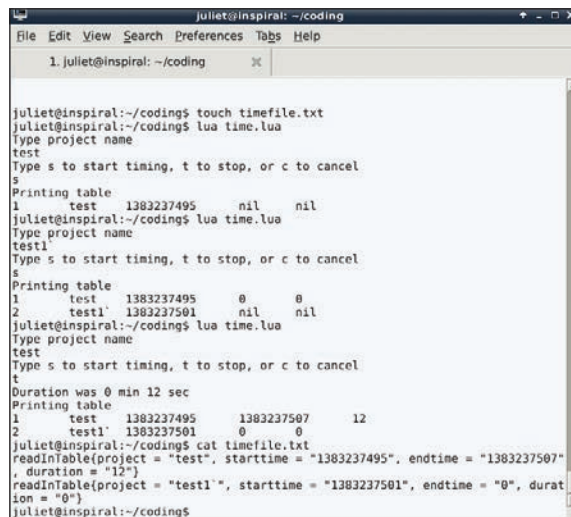
```
function serialiseTable()
    local f = assert(io.open(filename, "w"))
    for i, v in ipairs(timetable) do
        f:write(string.format(
            "readInTable{project = \"%s\", starttime = \"%d\", endtime = \"%d\", duration = \"%d\"}\n",
            v.project, v.starttime, v.endtime or 0, v.duration or 0))
    end
    f:close()
end
```

```
-- Главный код как раньше
outputTableToScreen()
serialiseTable()
```

Код открывания файла и записи в файл, а также итерации таблицы по большей части должен быть вам знаком, хотя строка немного неуклюжа (обратите внимание на экранированные кавычки). Единственное нововведение в этом коде — синтаксис `v.endtime or 0`. Это удобная конструкция Lua, которая читается так:

```
if not v.endtime then v.endtime = 0 end
```

Она нужна потому, что если у нас есть только имя проекта и время запуска, то `v.endtime` и `v.duration` не инициализированы и равны `nil`; а `string.format` не берет `nil`. Запустите программу и гляньте в выходной файл — там должны сохраниться новые значения.




Наконец, функция для добавления времени окончания задачи:

```
function tableRecordEnd(title)
    local timenow = os.time()
    for i, v in ipairs(timetable) do
        if v.project == title and tonumber(v.endtime) == 0 then
            d = timenow - v.starttime
            table.remove(timetable, i)
            table.insert(timetable, i, {project = v.project, starttime = v.starttime, endtime = timenow, duration = d})
            print(string.format("Продолжительность %d минут %d секунд", d/60, d % 60))
            break
        end
    end
end
dofile(filename)
-- Строки ввода как раньше
if input == "s" then
    tableRecordStart(project)
elseif input == "t" then
    tableRecordEnd(project)
else
    error("старт проекта отменен")
end
outputTableToScreen()
serialiseTable()
```


Функция `tableRecordEnd()` проходит по таблице в поисках строки с нужным именем проекта, для которого не указано время окончания. Обратите внимание, что при получении номера из `v.endtime` надо воспользоваться функцией `tonumber()`, так как номер считается из файла и поэтому представляет собой строку.

Найдя необходимую строку, ее нужно изменить. Но поскольку объекты в Lua представляют собой таблицы и обрабатываются через ссылки, нельзя просто изменить значения и ожидать, что они изменятся в «настоящей» таблице. Вместо этого нужно удалить всю строку и вставить новую на ее место. После этого мы сообщаем пользователю, сколько он работал над задачей, и все!

Как всегда, можно сделать еще массу вещей — скажем, анализировать время, затрачиваемое на различные проекты, или более сложные информационные параметры. Одно из главных применений Lua в реальном мире — встраиваемые программы, и если вам это интересно, есть много способов продвинуться дальше или принять участие в любом из понравившихся проектов.

У Lua хорошая онлайн-документация и процветающее сообщество, и поддержка экспериментов с этим приятным и гибким языком вам обеспечена. 

▶ Запустите программу и посмотрите на результирующий файл. Если этого файла не существует, вы получите сообщение об ошибке от `dofile()` — в программу полезно добавить проверку наличия файла.

Скорая помощь 

Другой способ сделать файл исполняемым — добавить в него «шапку» (строку с «шалашом») `#!/path/to/luac` (`print("Hello World")`). После этого его можно запустить командой `./hello.lua`.

Squeak: Введение

Squeak, свободная версия Smalltalk с собственной VM, отличается исключительно переносимостью, очень сильно объектно-ориентирована и предоставляет совершенно непохожую среду программирования для **Джульетты Кемп**.



Наш эксперт

Джульетту Кемп сначала отпугнула вся эта графика, но потом захватила простота создания рабочего кода.

Squeak — это реализация с открытым кодом Smalltalk, языка, которому уже 35 лет и который является первым действительно объектно-ориентированным языком. Все в Smalltalk является объектом, и все делается посредством отправки сообщений объектам. Squeak предлагает среду разработки с высокой самодостаточностью: все в вашем проекте Squeak также написано на Smalltalk и может быть обследовано, в том числе графические объекты и сама виртуальная машина (VM) Squeak. В частности, это очень удобно для образовательных учреждений, хотя и для многих других проектов тоже. Читайте дальше, чтобы познакомиться с ней в реализации короткой игры.

Приступаем

Вы можете установить Squeak либо через свою систему пакетов (*squeak-vm* в Debian/Ubuntu), или как один ZIP-файл с сайта. Если вы делаете загрузку через систему пакетов, вам также придется скачать образ Squeak с сайта Squeak. В качестве альтернативы, если вы используете скачанный с сайта ZIP, просто распакуйте файл SH, сделайте **cd** в распакованную директорию и введите **./squeak.sh**.

Когда вы запустите Squeak, перед вами появится большое окно со множеством маленьких окошечек, содержащих базовую информацию по Squeak. Первое, что вы должны сделать — создать новый проект. Выберите **Projects > New Project > New MorphicProject**, и вы получите новый проект — окно Squeak со всеми его подокнами и разными вкладками по краям (см. *Среда Squeak*, стр. 89, чтобы узнать, как запускать разные части окна проекта Squeak).

Хотя Smalltalk включает опцию для проектов MVC (набор инструментов MVC из Smalltalk-80), сама часть MVC выглядит устаревшей и больше не поддерживается. Morphic делает все то же самое, и даже больше, и выглядит симпатичнее, так что лучше

использовать его. Morphic также доступен на других языках (например, Self), и разработан для создания GUI.

Чтобы приступить к созданию самого кода, щелкните по фону окна проекта, и перед вами появится меню **World**. Выберите в нем **Workspace**, чтобы увидеть окно рабочего пространства (где вы будете создавать код) и **Transcript**, чтобы увидеть окно расшифровки (где будут отображаться результаты и ошибки кода). В качестве альтернативы, можете щелкнуть по вкладке **Tools** в дальнем правом углу, и перетащить оттуда окна **Workspace** и **Transcript**.

В окне **Workspace** введите:

```
Transcript show: 'Hello World'.
```

Вы должны использовать одиночные, а не двойные кавычки. (Двойные кавычки в Squeak отображают комментарии, и кавычки нужны с обеих сторон комментария.) Выделите, дважды щелкните и выберите **Do It** [Сделать] для запуска, и вы увидите, как 'Hello World' появится в окне **Transcript**. Точка в конце строки, завершающая утверждение в коде, для одной строки необходимой не является, но у нас строк будет несколько, и мы ее поставили.

В качестве альтернативы, есть еще более быстрая опция — введите в **Workspace**

```
'Hello World'
```

Затем выделите это, дважды щелкните и выберите **print**. Hello World появится непосредственно в **Workspace** — **print** также работает как вычислитель. Попробуйте ввести математическое выражение, например, **65*30**, выделите и выполните **print**, чтобы увидеть результат в **Workspace**.

Грамматика Squeak

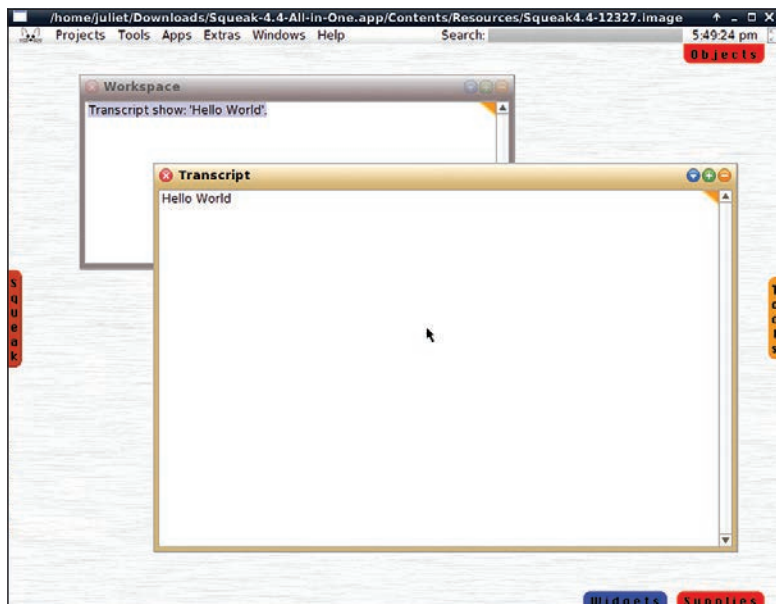
Возможность оценки небольших фрагментов кода может быть весьма удобна, но строка **Transcript show** демонстрирует грамматику Squeak немного лучше. Squeak — это полностью объектно-ориентированный язык: все в Squeak является объектом. Поэтому базовое «утверждение» кода Squeak — **object message**. Здесь **Transcript** — это объект (окно **Transcript**), а **show: 'Hello World'** — сообщение [message]. В данном случае сообщение состоит из метода (**show**) и аргумента, но подробности будут изменяться в зависимости от того, что вы делаете.

Squeak настолько хорош в графике, что мы также можем с легкостью создать кнопку Hello World:

```
button := SimpleButtonMorph new.
button color: Color green.
button label: 'Say hello'.
button addMouseUpActionWith: 'Transcript show: 'Hello World''.
button openInWorld.
```

Код создает новый объект **SimpleButtonMorph**, **button**, закрашивает его зеленым и присваивает ему ярлык. Кнопка **addMouseUpActionWith** делает то, что заявляет: добавляет действие для выполнения, когда определяется **MouseUp** (т.е., после того, как пользователь закончил нажимать на кнопку, что нормально для щелчков мышью). Действие — наша более ранняя строка **Transcript**. На этот раз строке нужны двойные кавычки (однако это две одиночных кавычки, а не символ двойных кавычек). И, наконец, **openInWorld** показывает кнопку в Squeak World, т.е. внутри окна Squeak.

➤ **Окна Workspace и Transcript со знаменитой программой — всеобщим первенцем.**



Выделите код и выполните его, и вы увидите кнопку. Можете также заставить кнопку изменить свой ярлык:

```
button addMouseUpActionWith: 'self label: 'Hello World!' ;'
```

Написание программы

Давайте перейдем к чему-то более основательному. Напишем небольшую программу для игры в Tic-Tac-Toe (она же — крестики-нолики). Мы создадим игровое поле 3×3, на котором при щелчке мышью будет появляться O или X. У нас будет класс для поля и класс для ячеек, так что разберем, как Squeak использует классы.

Сначала создайте новый проект Squeak, чтобы у вас была чистая область для работы. Теперь добавим категорию классов, чтобы хранить в ней классы. Перейдите в Tools > System Browser, и вы увидите браузер, который покажет вам все классы Squeak. Щелкните правой кнопкой по левой панели, что покажет все категории, и выберите пункт Add. Дайте ему имя Tic-Tac-Toe и нажмите на Accept. Вы увидите новую категорию в списке, хотя в ней пока нет никаких членов.

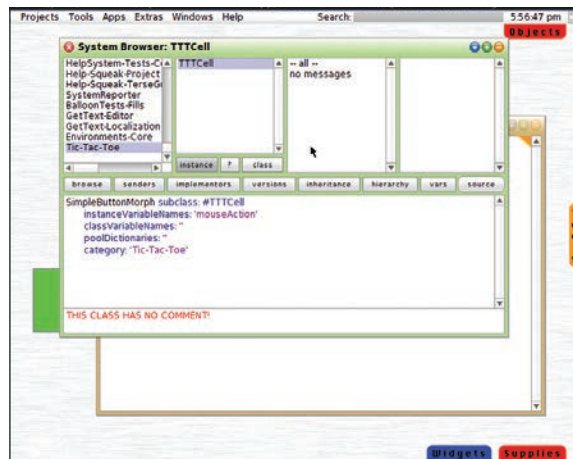
Зато в ней есть шаблон в нижней панели, которая является панелью редактирования классов. Это образец для метода, который посылает сообщение subclass в класс Object, с параметром #NameOfSubclass — чтобы класс Object создал новый подкласс сам по себе, с указанным именем. (Вы также можете описать это как вызов метода subclass класса Object; но в Smalltalk мы говорим об объектах и сообщениях, а не о методах).

Отредактируем класс ячеек нашей игры, чтобы он выглядел таким образом:

```
SimpleButtonMorph subclass: #TTTCell
instanceVariableNames: 'mouseAction'
classVariableNames: ''
poolDictionaries: ''
category: 'Tic-Tac-Toe'
```

Итак, теперь, вместо того, чтобы создавать подкласс Object, мы создаем подкласс SimpleButtonMorph (основная кнопка для нажатия) под названием TTTCell. Мы также добавляем переменную экземпляра mouseAction (которая существует в каждом экземпляре класса) в класс, поэтому мы можем действовать в ячейке в зависимости от того, что делает мышь.

Чтобы сохранить эти изменения, щелкните правой кнопкой и щелкните по Accept. Вы увидите, как внизу появится красное сообщение, предупреждающее вас, что по этому классу нет комментариев. Чтобы добавить короткий комментарий, щелкните



по кнопке ? между кнопками экземпляра и класса в панели System Browser, напишите комментарий и снова выберите Accept.

Далее мы можем написать настоящие методы. Первый, который вам нужен, это метод инициализации, который настраивает новую пустую оболочку. Выберите TTTCell в списке классов, затем все в списке протоколов за ним. Вы получите метод образца в панели редактирования ниже. Замените его таким кодом:

```
initialize
super initialize.
self label: ' '.
self borderWidth: 5.
self width: 50.
self height: 50.
color := Color paleBlue.
self useSquareCorners.
```

Этот метод начинается вызовом метода суперкласса initialize, который задаст любые разумные настройки по умолчанию, которые есть в суперклассе. Затем мы задаем ярлык ячейки (на пустую строку — и снова здесь две одиночных кавычки, а не двойные кавычки), его границу, размер, цвет и т. д. Большая часть всего этого самоочевидна. По существу, каждая строка является сообщением, отправленным объекту. Вы можете попробовать это для себя, открыв Workspace и введя

```
t := TTTCell new.
t openInWorld.
```

» Добавление первого класса. Обратите внимание на красное предупреждение внизу — я еще не добавляла комментарий. (Плохой кодер, не заслужила печеньки.)

Скорая помощь

Squeak — это реализация и среда, а Smalltalk — это язык, хотя в обычном использовании они пересекаются, что и будет отражено во всей остальной части урока.

Среда Squeak

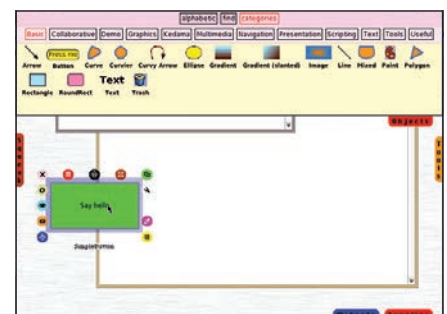
Если вы привыкли к программированию в командной строке и стандартном редакторе, все это выглядит слегка необычно. (Иными словами: меня оно озадачило.)

Сначала кое-что о мышях. Squeak предполагает наличие трех кнопок мыши. Поскольку это справедливо не для каждой мыши, то расположения меню в разных системах отличаются. Левая кнопка мыши постоянно используется для выбора, правая кнопка обычно выводит меню опций, а средняя кнопка (или Alt-левая стрелка или Ctrl-левая стрелка) выведет «ореол» кнопок вокруг объекта. Однако эти две последних (которые также именуются «желтая» и «голубая») иногда меняются местами; так что постарайтесь экспериментально установить, где какая в вашей системе.

Когда вы откроете проект Squeak, вокруг экрана будет несколько вкладок. Objects покажут вам

ваши объекты (например, кнопки и стрелки), которые можно перетаскать на экран; Tools показывает полезные объекты; Widgets и Supplies — другие полезные подмножества Objects. Любые из них можно перетаскать на экран, чтобы приписать им значение. Затем можете щелкнуть по средней кнопке или правой кнопке, чтобы вывести разные опции. Правый щелчок и inspect или explore покажут вам информацию об объекте, с окном кода внизу, где вы сможете попробовать код. Это руководство будет сосредоточено на кодировании с использованием System Browser, который позволяет добавлять классы и код таким способом, который будет знаком вам, если у вас есть опыт программирования; однако можно сделать очень многое с помощью графической системы Morphic (см. *Squeak... Детям, детям!* на стр. 91). На него определенно стоит потратить некоторое

время, с целью увидеть, что вы можете сделать с его помощью.



» У зеленой кнопки только что изменили размер с помощью желтой кнопки в ореоле; а окно Objects готово предоставить объекты.

Скорая помощь



Для изучения любого из классов Squeak можно использовать System Browser. Щелкните правой кнопкой и выберите 'find' в верхнем левом окне, чтобы найти класс, и используйте кнопку ?, чтобы просмотреть его документацию.

Выделите его и выберите Do It, чтобы создать новую ячейку. Вы также можете ввести **TTTCell new**, выбрать Inspect It, затем ввести **self openInWorld**. в панели внизу окна Inspection и выбрать применение к этому Do It. Окно Inspection также позволит вам проверять разные прочие аспекты объекта, так что неплохо потратить на него некоторое время.

Игровое поле

Наша ячейка пока что ничего не делает, но мы ее временно оставим и перейдем к игровому полю. Для создания класса игрового поля щелкните по категории класса Tic-Tac-Toe, и вы получите еще один экземпляр подкласса. Теперь отредактируйте его, чтобы он принял следующий вид:

```
BorderedMorph subclass: #TTTBoard
instanceVariableNames: ''
classVariableNames: ''
poolDictionaries: ''
category: 'Tic-Tac-Toe'
```

BorderedMorph — это именно морф (графический объект), у которого есть граница. Примите этот метод [Ассерт], затем создайте метод initialise:

```
initialize
| singleCell width height n |
super initialize.
n := self cellsPerSide.
singleCell := TTTCell new.
width := singleCell width.
height := singleCell height.
self bounds: (10@10 extent: ((width*n)@(height*n)) + (2 * self
borderWidth)).
cellGrid := Matrix new: n tabulate: [: i : j | self newCellAt: i at: j].
```

На сей раз при выборе Ассерт Squeak начнет жаловаться. Первым, на что он пожалуется, будет **cellsPerSide**; подтвердите, и мы напишем ее через минуту. Второе — **cellGrid**; для него выберите Declare Instance, поскольку это будет переменная экземпляра. И, наконец, **newCellAt** — снова подтвердите, и мы скоро ее напишем.

Первая строка этого метода объявляет четыре временных переменных. Это хорошая практика — объявлять переменные, которые существуют только в методе, временными, чтобы не загромождать проект Squeak ненужными глобальными переменными и сократить вероятность случайной перезаписи переменных.

Сохранение и выход

Вроде бы есть несколько опций сохранения, но у меня получилось заставить работать только одну. Все мои попытки щелчка по Projects > Save Project заканчивались зависанием. Нажатие на вкладку Squeak и Save, или меню Squeak и Save, или меню Squeak, Quit, и ответ Yes на вопрос о сохранении, работали отлично; когда я заново открыла Squeak, мой код по-прежнему был там.

Вы также можете сохранить свой проект как особый образ, но у скрипта **squeak.sh** образ запуска с жесткой кодировкой, так что вам придется его отредактировать, написать свой скрипт или использовать этот код, чтобы запустить Squeak:

```
exec Contents/Linux-i686/bin/squeak my.image
```

Если вы потеряли код, при новом запуске он весь должен быть в Recover Changes, поскольку Squeak записывает все. Перейдите в Extras > Recover Changes, выделите нужные изменения, затем щелкните правой кнопкой и выберите файл.

Затем определим все эти временные переменные. Мы настроим границы игрового поля (оно кажется долго, но на самом деле выдает ожидаемый размер игрового поля на основании размера ячеек). И, наконец, мы создадим матрицу ячеек с правильным количеством ячеек.

Однако самая сложная строка — последняя. **cellGrid** (помните, мы определяли ее как переменную экземпляра, так что она доступна во всем классе) — это место, где мы будем хранить сетку ячеек. Мы создаем ее с помощью **Matrix>>new:tabulate:**. Это метод **new** класса Matrix. Нотация **Class>>method** используется, когда мы говорим о коде Squeak, хотя в самом коде она не применяется. Однако что здесь делает **tabulate**? В Squeak вы узнаете, сколько аргументов в методе, благодаря количеству двоеточий в его имени. Здесь два двоеточия означают два аргумента. Первый дает столбцы и строки Matrix, а второй указывает, какой объект находится в каждой (i, j)-й позиции матрицы. Этот последний использует сообщение **newCellAt**, поэтому каждая (i, j)-я ячейка состоит из того, что возвращено **newCellAt: i at: j**.

(Спасибо авторам превосходного *Squeak By Example* за части этого метода и за другие методы, используемые на нашем уроке).

Другие методы

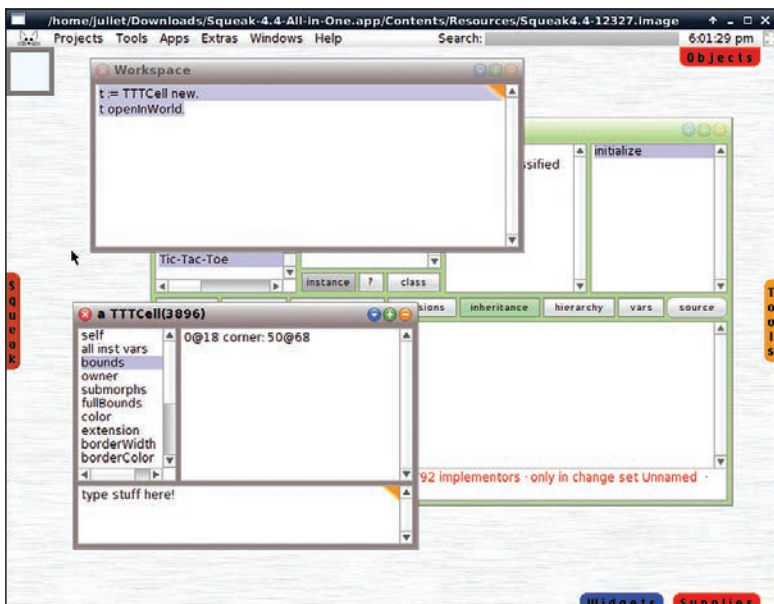
Все это означает, что нам нужны и другие методы. Щелкните по имени класса, и вы получите еще один образец метода. У нас есть два метода для написания:

```
cellsPerSide
"Number of cells on each side of the board"
^3

newCellAt: i at: j
"Create a cell at position i,j and add it to the screen"
| cell origin |
cell := TTTCell new.
origin := self innerBounds origin.
self addMorph: cell.
cell position: i - 1 * cell width @ (j - 1 * cell height) + origin.
cell
mouseAction: [self playMove: i at: j].
^cell
```

cellsPerSide просто возвращает 3 (оператор ^ означает «вернуть эту величину»). **newCellAt** делает то, что указано: создает новую TTTCell и определяет **origin**, т.е. начало координат внутренней границы TTTBoard. (Если вы считаете игровое поле решеткой, то это верхний левый внутренний угол.) Этот метод унаследован от суперкласса BoundedMorph. Затем мы рассчитываем и устанавливаем положение ячейки, установив, насколько далеко она

➤ TTTCell сверху, в верхнем левом углу. Ниже вы также можете видеть ее inspector, а за ними — System Browser.



расположена в поле и добавляя ее в **origin**, чтобы поместить в окно Squeak.

Важная часть — предпоследняя строка. Она устанавливает, что должна делать ячейка в ответ на действия мыши; примените метод `TTTBoard>>playMove:at:`. Это базовая логика игры, и мы напишем ее через минуту. Наконец, мы возвращаем вновь созданную ячейку.

Чтобы выполнить всю эту работу, надо написать еще несколько методов. Следующий — `TTTBoard>>playMove:at:`:

```
playMove: i at: j
(cellGrid at: i at: j)
  move: player.
self switchPlayer.
```

Однако если вы его приняли, Squeak начнет жаловаться, что он не знает о **player**. Добавьте его как экземпляр, и через минуту мы им займемся; **switchPlayer** просто подтвердите, и вскоре мы напишем и этот метод тоже. **cellGrid** — это матрица, так что она наследует методы матрицы — в частности, метод **at:at:** для определения нужной ячейки в матрице. Получив эту ячейку, мы применяем метод **move** с аргументом **player**; затем переключаем **player**.

Далее нам надо, чтобы **player** имел значение, показывающее, какой игрок в данный момент делает ход, X или O. X всегда начинает, так что мы просто добавляем строку к **initialize**:

```
initialize
super initialize.
player := 'x'.
“остальное как выше”
Нам еще нужно switchPlayer:
switchPlayer
player = 'x'
ifTrue: [player := 'o']
ifFalse: [player := 'x']
```

Первая строка оценивается как истинная или ложная (Булево значение). Если она истинна, выполняется строка **ifTrue**, и **player** настроен на O (ход O). Если ложная, то она настраивается на X. Учтите, что проверка ошибок за вас не выполняется, а значит, вам, возможно, понадобится что-то добавить, чтобы определить, не настроена ли она на нечто странное.

Вот и все по классу **TTTBoard**. Однако у нас по-прежнему есть несколько методов **TTTCell** для написания и обработки нажатий на мышшь.

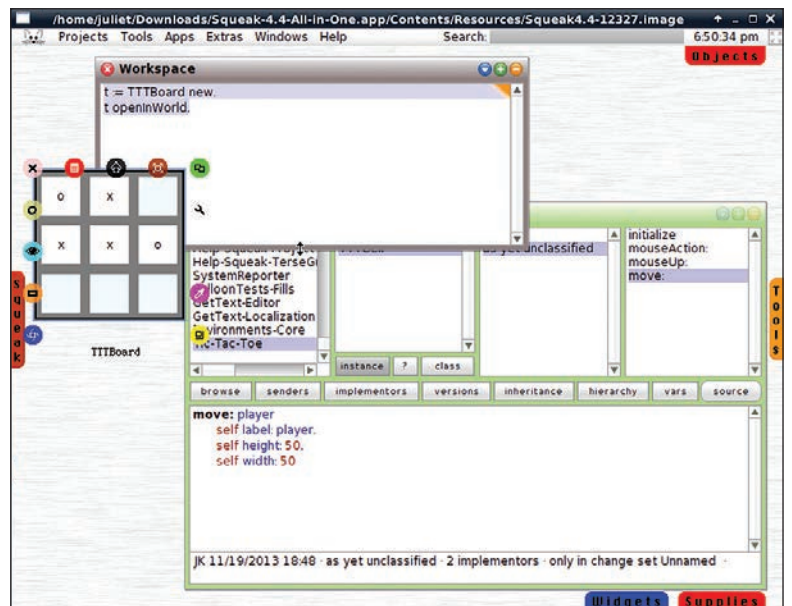
Действия мыши

Вернемся к **TTTCell**, чтобы поработать с действиями мыши и этим методом **move**.

```
mouseAction: aBlock
^ mouseAction := aBlock.
mouseUp: anEvent
mouseAction value.
```

Метод **mouseAction:** — это то, что вызывается на ячейку, когда по ней щелкают, из метода `TTTBoard>>newCellAt:at:`. Это также переменная экземпляра, если вы оглянетесь на код, использованный для настройки класса **TTTCell**. Тогда метод **mouseAction:** — это просто метод настройки, и он настраивает переменную экземпляра **mouseAction** на принятый в нее аргумент. А затем возвращает его. Итак, метод setter Smalltalk — это **variable**; и метод getter — это **variable** (обратите внимание: здесь нет двоеточия), вместо **setVariable** и **getVariable**, как в других языках. Эти методы называются методами **accessor**, и при категоризации методов их следует помещать в протокол **accessing**.

mouseUp: определяет, что происходит, когда пользовательский интерфейс определяет событие **mouseUp**. Здесь мы отправляем сообщение **value** в **mouseAction**. Этот метод был настроен в `TTTBoard>>newCellAt:at:` как `[self playMove: i at: j]` (то есть



► Игра в действии! Вы можете выделить ее, и вокруг появится ореол, чтобы **move/inspect/** и т. д. Вы также видите **System Browser** с кодом внизу.

`TTTBoard>>playMove: i at: j)`. Сообщение **value** дает команду Squeak оценить это выражение, чтобы исполнить **playMove:at:**.


`TTTBoard>>playMove` вызывает `TTTCell>>move`, и это конечный метод, который нам предстоит написать:

```
move: player
self label: player.
self height: 50.
self width: 50.
```

Это настраивает аргумент **player** в качестве метки ячейки. Затем он изменяет размер ячейки, поскольку настройка метки изменяет размер ячейки в соответствии с размером метки, что нам ни к чему. Для большей портативности и лучшей работы было бы неплохо, чтобы **height** и **width** были переменными экземпляра.

Игра готова! Теперь запустите ее и посмотрите, что произойдет. Как и всегда, вы можете сделать еще многое, чтобы улучшить эту игру, так что вперед — посмотрите, что еще вы сможете сделать.

Squeak был для меня неким отклонением от нормы, поскольку он сильно ориентирован на графику, а я склоняюсь к программированию в командной строке. Но это отличный язык — и отличная среда, если в ней разобраться. Было бы также интересно попробовать больше графических виджетов **drag-and-drop** для создания кода. Если вы хотите исследовать его дальше, вам поможет отличная книга *Squeak By Example*, она предлагает очень подробное введение в Squeak и свободно доступна онлайн. **LXF**

Скорая помощь 

Чтобы создать свой метод достаточно, вы можете щелкнуть правой кнопкой, щелкнуть `prettyprint`. Можете также соединить **Shift**-правую кнопку, чтобы сразу перейти в меню `more`.

Squeak... Детям, детям!

Один из привлекательных аспектов Squeak — то, что он очень доступен для детей. Даже те дети, которые еще не готовы попробовать программирование в Smalltalk, могут рисовать в Squeak и перетаскивать виджеты и команды скриптов.

Подобное кодирование позволяет создавать и соединять скрипты без необходимости что-то печатать, и это дает детям большую свободу в экспериментах с программированием.

Если вам это интересно, система **Etoys** (www.squeakland.org), предназначенная специально для детей и активно используемая в проекте **One Laptop Per Child** [Ноутбук — каждому ребенку], является частью Squeak. Похожая опция, основанная на Squeak/Smalltalk и разработанная MIT — **Scratch**, и ее можно использовать онлайн. Squeak, Etoys и Scratch дают детям отличную возможность начать программировать, а не просто потреблять программы.

Erlang: Тесты программ

Андрей Ушаков начинает цикл, посвященный качеству нашего кода и программ, создаваемых на любимом языке Erlang.



Наши
эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

Хороший способ улучшить качество кода — это проверить его работоспособность согласно некоторому набору тестов. Чем мы и займемся в данной серии уроков.

Что такое тестирование? Это процесс проверки работоспособности созданного программного обеспечения (некоторого продукта) и соответствия этого продукта заданным требованиям. Допустим, мы занимаемся разработкой такого программного продукта, как калькулятор. Тогда требования будут определять поддерживаемые нашим калькулятором функции: обычный это калькулятор (с поддержкой только четырех арифметических операций) или инженерный (с поддержкой тригонометрии и т. д.), должны ли мы поддерживать работу с системами счисления с основаниями, отличными от 10, и т. д. При проверке наличия в нашем калькуляторе тех или иных функций (например, функции `sin` для инженерного калькулятора) мы проверяем соответствие созданного нами калькулятора требованиям. А проверка, что $2+2$ будет 4 — это проверка работоспособности нашего калькулятора.

Давайте бегло рассмотрим, как выполняется тестирование программного продукта. Обычно для этого применяются т. н. пользовательские сценарии: когда повторяются действия пользователей для достижения того или иного результата. При наличии достаточного количества пользовательских сценариев, мы можем проверить как выполнение всех требований к нашему продукту, так и правильность его работы (хотя осуществить такую проверку не всегда просто). Давайте вернемся к нашему примеру с калькулятором. Так, действия пользователя, состоящие из ввода некоторого числа и нажатия на кнопку `sin` (с последующей проверкой полученного результата), служат для проверки поддерживаемых калькулятором функций (для проверки соответствия требованиям). А действия пользователя, состоящие из ввода числа 2, нажатия на кнопку `+`, ввода числа 2, нажатия на кнопку `=` и проверки полученного результата (который, естественно, должен быть 4), служат для проверки работоспособности нашего калькулятора. Очевидно, что нет особой разницы между двумя этими сценариями: они оба состоят из некоторых действий пользователя и проверки получаемого результата. В более сложных случаях мы также должны подготовить некий набор исходных данных для того, чтобы получать один и тот же результат при одних и тех же действиях пользователя (но это не всегда возможно). Поэтому для проведения тестирования абсолютно неважно, откуда появились сценарии для тестирования: из требований, из проверки работоспособности или при воспроизведении той или иной ошибки.

У многих читателей может возникнуть вопрос: а зачем разработчикам вникать в процесс тестирования, если заниматься тестированием должны специальные люди (QA-специалисты)? Да, при правильной постановке процесса разработки программисты не занимаются тестированием всего продукта согласно набору пользовательских сценариев. Но при этом они все же должны принимать участие в тестировании создаваемого ими программного

обеспечения. Как может разработчик принять в этом участие? Ответ очевиден: создать тесты на разрабатываемый им код, компоненты, подсистемы, библиотеки и т. д. При этом тестирование, которое осуществляет разработчик (при помощи написания тестов), называют автоматизированным тестированием, в отличие от ручного тестирования приложения, которое осуществляет QA-специалист (хотя и QA-специалисты могут разрабатывать тесты для автоматизированного тестирования).

Вполне закономерен вопрос о плюсах и минусах такого подхода к разработке. Начнем с тех преимуществ, которые он нам дает. Главное его преимущество в том, что все внутренние составляющие нашего приложения (компоненты, подсистемы и т. д.) оказываются протестированы (естественно, при 100% покрытии тестами исходного кода). Вернемся снова к пользовательским сценариям работы: каждый сценарий — это та или иная комбинация возможных входных данных для всех компонентов, из которых состоит наше приложение. Если мы хотим полностью протестировать наше приложение, то должны составить все возможные комбинации входных данных для всех компонентов нашего приложения, после чего каждой такой комбинации сопоставить некоторый пользовательский сценарий работы. Очевидно, что число таких комбинаций будет настолько велико, что подобную работу будет невозможно выполнить. Поэтому полностью протестировать наше приложение невозможно: можно проверить лишь основные сценарии работы и отсутствие известных ошибок (обычно сценарий воспроизведения ошибки становится сценарием проверки ее исправления).

С другой стороны, тестируя внутренние компоненты приложения по отдельности, мы сможем проверить гораздо больше возможных сценариев использования компонентов (и по входным данным, и по манипуляциям с этими данными), т. е. протестировать больше аспектов работы нашего приложения. Конечно, наличие тестов на компоненты не означает, что нет нужды в тестировании приложения согласно пользовательским сценариям; правильная работы отдельных компонентов не гарантирует правильности работы приложения в целом. Наличие тестов на компоненты нашего приложения означает, что можно безболезненно заниматься развитием и сопровождением кода (добавлять новые функции, исправлять ошибки в имеющемся коде, выполнять рефакторинг).

Вполне возможна ситуация, что в процессе работы над кодом тесты перестанут проходить. Значит, у нас либо тесты перестали соответствовать действительности, либо мы что-то добавили некорректно, в результате чего наша реализация перестала удовлетворять проверкам в тестах. В первом случае мы изменяем тесты — таким образом, чтобы они соответствовали действительности. Во втором случае мы исправляем наши изменения в коде, чтобы они снова начали удовлетворять всем проверкам во всех тестах. Нет ничего страшного, если во время работы над задачей тесты перестанут проходить: достаточно, чтобы они проходили по окончании работы над задачей.

Разработка тестов для существующего кода часто приводит к пониманию того, что код не очень удобен для написания тестов к нему. После этого нередко приходится менять исходный код. Обычно проблемы с исходным кодом заключаются в сильной связности между его частями. Для решения этой проблемы создают интерфейсную часть и реализацию для некоторой функциональности, чтобы взаимодействие с этой функциональностью происходило через интерфейс. Это приводит к уменьшению связности кода, а также к тому, что мы в будущем при желании сможем изменить реализацию функциональности, не затрагивая использующий ее код. Как итог, разработка тестов часто вынуждает нас к некоторой декомпозиции исходного кода, что приводит к уменьшению его связности и увеличению гибкости дальнейшего использования. А это улучшает общую архитектуру системы.

С разработкой тестов связан еще один немаловажный момент: раннее обнаружение ошибок. Если ошибку обнаруживает QA-специалист, то он ищет стабильный сценарий ее воспроизведения, после чего регистрирует ее в баг-трекере. В какой-то момент разработчик берется за ее исправление. Для этого он воспроизводит ошибку у себя, после чего ищет в коде причину возникновения ошибки и исправляет ее. На все эти шаги может уйти достаточно много времени и сил. Если же разработчик при помощи тестов находит ошибку в коде, то он просто исправляет ее — и все.

И еще один плюс наличия тестов: они являются хорошей документацией по использованию кода и по внутреннему его поведению. С документацией на код очень часто бывает много проблем: ее могут забыть актуализировать; ее может писать отнюдь не разработчик, и в результате — как работать с кодом, совсем не понятно; в конце концов, она может содержать ошибки, которые никто не заметил. Тесты на код, конечно, не могут заменить документацию (особенно хорошую), но, тем не менее, они неплохо ее дополняют. К тому же тесты обычно всегда актуализированы относительно кода, их пишут разработчики (во многих случаях это тот же человек, который писал код) и в них меньше вероятность появления ошибки, так как при наличии ошибок код, скорее всего, не будет удовлетворять проверкам в тестах, что будет поводом к проверке самих тестов.

Рассмотрев плюсы написания тестов, определимся с ценой, которую мы вынуждены платить при таком подходе к разработке.

Во-первых, при таком подходе время разработки увеличивается в несколько раз (обычно в 2–3 раза). Это связано с тем, что сложность и объем кода тестов обычно сопоставимы со сложностью и объемом соответствующего им основного кода. Учтем, что при таком подходе в несколько раз увеличивается и общий объем кода, несколько усложняя сопровождение и дальнейшую модификацию кода. Поэтому, когда есть потребность очень быстро получить код (например, по пилотному проекту или макету приложения), тестов совсем не пишут либо пишут их по минимуму (например, общий интеграционный тест на все приложение).

Во-вторых, так как время жизни тестов сопоставимо со временем жизни кода, то к написанию тестов следует подходить точно так же, как и к написанию кода. Это означает, что нужно продумывать их архитектуру, возможное повторное использование каких-то частей тестов, избегать дублирования кода и т. п. Что

опять же увеличивает время, необходимое на написание тестов существующего кода.

В-третьих, написание хороших тестов, покрывающих все возможные сценарии выполнения кода со всеми возможными вариантами данных, достаточно сложно и нетривиально.

И, наконец, в-четвертых — это человеческий фактор. Как показывает практика, для многих разработчиков написание тестов к своему коду гораздо менее интересно, чем написание самого кода. Поэтому бывают ситуации, когда разработчики относятся к написанию тестов формально и спустя рукава. Понятно, что ценность тестов, созданных при таком отношении, сильно ниже, чем могла бы быть. Еще вариант, связанный с человеческим фактором — это когда написанием тестов к коду занимается специальный человек (часто — новичок в разработке), а не разработчик, который пишет сам код. В итоге тесты получаются не очень хорошего качества и, возможно, с неполным покрытием кода. Связано это с тем, что для написания качественных тестов необходимо понимать исходный код, а стороннему человеку обычно сложнее вникнуть в код, нежели автору. Следует сказать, что подход с отдельным разработчиком для написания тестов на приложение бывает и оправданным: например, для написания интеграционных тестов или функциональных тестов на подсистемы.

Также следует сказать, что в некоторых ситуациях без написания тестов на существующий код никак не обойтись. Представьте такую ситуацию, что вашим продуктом является библиотека (причем неважно, для внутреннего использования или для внешнего мира) с некоторой реализованной функциональностью. Как ее можно протестировать без написания кода (в данном случае, тестов), только по набору пользовательских сценариев? Ответ очевиден: никак. Поэтому для тестирования такого продукта нам придется написать некоторый объем кода. При написании тестов нам понадобятся сценарии использования этой библиотеки. Только, в отличие от пользовательских сценариев, это будут не действия пользователей, а набор сценариев для выполнения всех возможных типичных задач.

Мы оценили плюсы и минусы написания тестов при разработке кода (надеюсь, что плюсы для вас перевесили минусы, и вы решили, что будете писать тесты на код, если уже не делаете этого). Рассмотрим более пристально, какие бывают виды тестов. Начнем мы с наиболее, пожалуй, популярного вида тестов, которые пишут разработчики: модульных тестов (они же unit-тесты). Это тесты для проверки отдельных единиц кода (процедур, функций, классов и т. д.), причем при проверке мы можем работать с проверяемыми единицами кода как с «белым ящиком» (об этом ниже). Модульные тесты никак не должны зависеть от окружения; им достаточно для работы только исходной единицы кода (и дополнительных библиотек для создания самих тестов). Так, например, если для успешного выполнения модульных тестов необходимо существование некоторого файла с некоторым содержанием в некоторой директории, то такой тест не может считаться модульным. Также, модульные тесты должны работать независимо друг от друга. Это означает, что результат выполнения любого теста должен быть всегда одним и тем же, вне зависимости от того как мы запускаем этот тест: одного или вместе

»

Разработка через тестирование

Разработка через тестирование [test-driven development, TDD] — это подход к разработке программного обеспечения, при котором сначала пишутся тесты, а потом уже код, который эти тесты проверяют. Весь созданный код (практически весь) при этом автоматически оказывается покрыт тестами.

При этом тесты определяют требования к коду и его дизайн. При таком подходе разработка основывается на повторении очень коротких циклов, состоящих из создания тестов, написания кода, удовлетворяющего этим тестам, и рефакторинга тестов и кода. Следует заметить, что для покрытия всего кода

тестами не обязательно разрабатывать программное обеспечение согласно этому подходу. Вполне можно сначала разработать код, а потом создать для него тесты; но нужно быть готовым к тому, что для создания тестов (в особенности модульных) уже написанный код необходимо будет изменить.

Непрерывная интеграция

Непрерывная интеграция [continuous integration] — это подход к разработке программного обеспечения, при котором сборка проекта осуществляется периодически, в автоматическом режиме (обычно при помощи специальных средств, таких как Hudson, Jenkins и т.д.). При этом обычно выполняются все существующие тесты в проекте. Если во время процесса сборки возникли какие-либо проблемы или какие-то из тестов выполнены с ошибкой, это является сигналом, что с проектом что-то не в порядке. Поскольку сборка

проекта осуществляется периодически (обычно каждую ночь), то о проблемах со сборкой разработчики узнают практически сразу же (и сразу же все эти проблемы правят), а не на финальной стадии разработки проекта (когда необходимо собрать проект для заказчика/конечного пользователя). Еще одним большим плюсом такого подхода является тот факт, что QA-специалисты могут приступить к проверке новой функциональности или исправлений ошибок сразу же, как только разработчик завершит свой труд.

с другими тестами, а также вне зависимости от порядка выполнения разных тестов. Такое требование также может привести к тому, что возникнет необходимость в изменении кода, что приведет, в конечном итоге, к уменьшению связности внутри основного кода. Наиболее правильный подход при написании модульных тестов — это если такие тесты пишут разработчики, которые и создали тестируемый код. Одно большое исключение из этого правила — это работа с унаследованным кодом, в котором не было тестов. В этом случае разработчик, прежде чем решать некоторые задачи, пишет модульные тесты (возможно, и не только их) к уже существующему коду, созданному его предшественником.

Следующий вид тестов, о которых мы поговорим — это функциональные тесты, т.е. такие, которые проверяют некоторую функциональность всей системы или одной из ее подсистем (или некоторого компонента). При тестировании функциональности мы работаем со всей системой или некоторой подсистемой как с «черным ящиком» (об этом ниже).

Функциональные тесты работают с более крупными объектами, чем модульные. Поэтому создание функциональных тестов часто проще, чем модульных, т.к. мы работаем с абстракциями более высокого уровня. Но с другой стороны, достаточно часто для выполнения функциональных тестов необходимо сделать дополнительные действия по инициализации и очистке. Например, если мы тестируем подсистему работы с базой данных, то перед тестами необходимо эту базу создать, создать ее структуру и заполнить ее данными. Как и в случае модульных тестов, мы, конечно, желаем, чтобы результат выполнения каждого теста не зависел от результатов выполнения других тестов. Для достижения этого результата нам необходимо либо чистить базу перед выполнением каждого теста, либо заново создавать ее. Очевидно, что это усложняет инициализацию тестов и увеличивает время, потребное для их инициализации. А это приводит к увеличению времени прогона всех тестов. Поэтому модульным и функциональным тестам нужны разные сценарии запуска.

Обычно разработчик при работе над кодом периодически запускает на выполнение только модульные тесты (особенно перед сохранением результатов работы в системе контроля версий). А функциональные тесты (впрочем, как и модульные), обычно запускаются на выполнение на некотором выделенном сервере. За написание функциональных тестов не обязательно должен отвечать тот же человек, что и писал код. Часто бывает, что тестируемая подсистема или система настолько велика, что ее разрабатывают несколько программистов. При этом написание функциональных тестов может занимать достаточно большое количество времени. В такой ситуации вполне логично, чтобы за написание функциональных тестов отвечал отдельный разработчик (часто с более низкой квалификацией, по сравнению

с разработчиками основного кода). Хотя для небольших подсистем или систем такие тесты вполне может писать и разработчик данного программного обеспечения (или же в случае, когда с подсистемами работает всего один разработчик).

Из функциональных тестов можно выделить несколько подвидов тестов, к которым нужен особый подход. Начнем с интеграционных тестов. Это тесты на всю систему: при этом мы все подсистемы интегрируем в одну итоговую систему (как она будет выглядеть при реальном использовании). Очевидно, что развернуть такую систему перед тестами достаточно сложно. Если итоговая система обладает развитым пользовательским интерфейсом (на основе Web или некоторого оконного интерфейса), то ее крайне сложно протестировать без специальных средств. Например, для тестирования пользовательского интерфейса на основе Web применяются такие средства, как Selenium (см. <http://docs.seleniumhq.org/>). Эти средства эмулируют действия пользователя при работе с интерфейсом, что позволяет писать интеграционные тесты прямо на основе пользовательских сценариев использования системы. Причем такие тесты очень чувствительны даже к небольшим (иногда и незаметным для пользователя) изменениям в пользовательском интерфейсе: мелкое изменение способно привести к тому, что практически все тесты перестают выполняться. Если итоговая система не обладает развитым пользовательским интерфейсом, то ее тестирование гораздо проще и часто не требует специальных средств (как, например, в случае REST-сервисов). Но усилия на развертывание все равно придется затратить (а также усилия по обеспечению независимости результатов выполнения тестов).

Следующий вид тестов, который можно выделить — это тесты на производительность, нагрузочные и стресс-тесты. Все они служат для проверки работы системы и отдельных подсистем под нагрузкой. Тесты на производительность служат для измерения времени выполнения, как отдельных фрагментов кода, так и целых подсистем. Нагрузочные тесты служат для тестирования системы (и, иногда, некоторых подсистем) под ожидаемой рабочей нагрузкой (т.е. под нагрузкой, которая описана в требованиях к проекту). Стресс-тесты служат для тестирования системы (и, иногда, некоторых подсистем) под высокой и даже пиковой нагрузкой. Все эти тесты объединяет их повышенная сложность (закрывающаяся как в сложности развертывания окружения, так и в сложности самих тестов) и достаточно большое время выполнения.

А теперь взглянем более пристально на процесс написания тестов и на принципы, лежащие в основе этого процесса. Начнем с подхода к тестированию системы как «черного ящика». Как уже говорилось, данный подход используется при создании функциональных тестов на систему или на отдельные ее подсистемы.

Когда мы говорим о некоторой системе как о «черном ящике», это означает, что мы ничего не знаем о внутреннем устройстве данной системы. Все, что мы можем сделать с «черным ящиком» — это подать на вход одни данные и на выходе получить другие (здесь под входом и выходом мы понимаем любые точки для ввода и вывода данных: файлы, базы данных, некоторый набор API, логи, консоль и т.п.). При этом не гарантируется идемпотентность данной операции: т.е. если мы одни и те же данные подадим на вход два раза, то не факт, что во второй раз на выходе получим то же самое, что и в первый раз. В большинстве случаев, тестируемая нами подсистема является некоторым конечным автоматом: другими словами, у нее есть набор состояний, и при каждом взаимодействии с ней подсистема переходит из одного состояния в другое. Все возможные состояния и все возможные переходы мы можем получить из требований, ТЗ и прочей подобной документации к подсистеме (данная документация вполне может быть в некотором неформализованном виде).

Очевидно, что в итоге наши функциональные тесты (тестирование с точки зрения «черного ящика») будут выглядеть

следующим образом. Мы инициализируем подсистему, чтобы попасть в одно из ее состояний (не обязательно начальное). После этого при помощи одного или нескольких переходов мы переводим систему в некоторое другое состояние (необязательно заключительное). И, наконец, мы проверяем выходные данные состояния, в которое мы в итоге перешли.

Мы можем использовать подход к тестируемому объекту как к «черному ящику» и при модульном тестировании. Но такой подход далек от оптимального. Во-первых, на отдельные части кода достаточно редко (хотя на некоторые критичные части так делают) пишутся требования и/или нечто подобное. Соответственно, это приводит к тому, что построить конечный автомат становится не так уж и просто. Во-вторых, мы никак не пользуемся тем фактом, что у нас есть доступ к исходному коду тестируемого фрагмента. Поэтому создание модульных тестов на основе некоторых знаний об исходном коде тестируемых фрагментов является оптимальным.

Можно сразу же предложить очень прямолинейный подход для создания необходимого набора тестов: для тестируемого фрагмента взять все его входные параметры, построить все возможные комбинации допустимых значений для этих параметров, после чего для каждой такой комбинации написать отдельный тест. Однако стоит учесть следующий факт: существует достаточно мало типов данных, диапазон возможных значений которых конечен и достаточно мал. Так, например, у булевского типа данных всего 2 значения. Но с другой стороны, у строк в большинстве языков программирования множество возможных значений бесконечно, а у 32-разрядного целого числа множество возможных значений лежит в диапазоне от -2,147,483,648 до 2,147,483,647 (всего 4,294,967,296 значений). Поэтому в жизни такой очень прямолинейный подход практически не применим. Конечно, возможны ситуации, когда, например, мы тестируем функцию, которая принимает несколько входных параметров булевского типа; в этом случае мы можем составить все возможные комбинации входных параметров и для каждой комбинации создать свой тест. Но такие ситуации достаточно редки.

Давайте воспользуемся тем фактом, что у нас есть исходный код: при его анализе мы можем выделить все возможные пути выполнения кода (а также все возможные варианты вычисления выражений). Если мы создадим тесты таким образом, чтобы проверялись все эти возможные пути (т.е. каждый тест выполняет код по одному из возможных путей), то этого будет достаточно для проверки тестируемого фрагмента кода. При этом следует внимательно учесть следующий момент: каждому из возможных путей выполнения кода соответствует некоторое подмножество из множества всех возможных комбинаций входных параметров.

Понятно, что мы не можем взять все возможные комбинации входных параметров из такого подмножества. Поэтому обычно

поступают следующим образом. Обязательно берут некоторые осмысленные значения (например, для строк подключения к базе данных), берут граничные значения, если они существуют, и берут несколько любых произвольных значений из подмножества значений для входного параметра. После чего строят все возможные комбинации выбранных таким образом значений для всех входных параметров и проверяют в рамках одного теста для какого-либо пути выполнения исходного кода.

Но при написании модульных тестов может возникнуть следующая проблема: взаимодействие тестируемого кода с внешними подсистемами, такими как база данных, файловая подсистема и т.д. Если мы попробуем написать на такой код модульные тесты «в лоб», то, очевидно, ничего не получится. Причина в том, что для работы таких тестов необходимо сделать некоторую инициализацию внешних подсистем, от которых тестируемый код зависит. Как только мы добавим такую инициализацию в наши тесты, так сразу же наши тесты перестанут быть модульными.

Возникает известный русский вопрос: что делать? Ответ очевиден: наш код (который мы хотим тестировать) имеет взаимосвязь с конкретной внешней подсистемой, т.е. у нашего кода и у внешней подсистемы сильная связность. И первый шаг, который необходимо сделать — это разорвать такую взаимосвязь (уменьшить связность нашего кода). Для этого в коде связь с конкретной внешней подсистемой мы заменяем связью с некоторым интерфейсом, реализующим необходимые нам операции. Для работы после такой модификации необходимо передавать конкретную реализацию для введенного интерфейса либо при инициализации кода, либо при его использовании. Такая модификация, помимо уменьшения связности, дает нам возможность заменить одну конкретную внешнюю подсистему на другую (например, мы можем заменить работу с файловой системой на работу с REST-сервисом).

Второй шаг, который необходимо сделать для написания модульных тестов — это создать некоторый объект, имитирующий некоторую реальную внешнюю подсистему (mock-объект). Этот объект не будет требовать инициализации и может проверять порядок вызова своих методов и передаваемые параметры. Таким образом мы, помимо создания заглушки для реальной внешней подсистемы, можем тестировать то, как наш код взаимодействует с внешней подсистемой (т.е. поведение нашего кода). Но об этом мы поговорим очень подробно в одной из следующих статей.

В этом месяце мы рассмотрели концептуальное представление о том, что такое тестирование кода разработчиком. Мы узнали о принципах и подходах, о проблемах и их решении, о преимуществах и цене при написании тестов на код. Все, что было здесь сказано, применимо к любому языку программирования, любой платформе и операционной системе. О конкретике, связанной с языком Erlang, мы поговорим в следующей раз. **LXF**

Скрытые зависимости и тесты

При создании тестов мы желаем, чтобы результат выполнения каждого теста никак не зависел от результата выполнения других тестов. Если результат выполнения тестов зависит от порядка их выполнения, это делает подобные тесты, по большому счету, бесполезными. Когда подобные зависимости между тестами возникают, необходимо найти причину этой зависимости и ее устранить. Модульные тесты не зависят от внешнего окружения (файловой системы, баз данных и т.п.), поэтому зависимость между модульными тестами говорит о том, что между вызовами тестов сохраняется состояние

(например, в глобальной переменной или в статическом поле класса). Это говорит о высокой связности внутри кода.

Если проблема в коде теста, то достаточно переписать такие тесты. Если же проблема в тестируемом коде, то, возможно, потребуется переработка архитектуры кода, иногда затрагивающая весь проект. С другой стороны, выявление и устранение подобной зависимости делает код проще и снижает вероятность возникновения «side-эффектов». Функциональные тесты, в отличие от модульных тестов, зависят от окружения (они специально создаются

так, чтобы проверить работу подсистем в реальном окружении). Поэтому между такими тестами возможны зависимости и через код (как в случае с модульными тестами), и через данные (через внешнее окружение). В случае зависимости тестов через код мы поступаем точно так же, как и в случае модульных тестов. Зависимость тестов через данные говорит нам о том, что при инициализации внешних подсистем не все было инициализировано (или почищено). Это означает, что необходима доработка кода инициализации (или очистки) для функциональных тестов.

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru, и мы найдем ответ.

В этом месяце мы ответим на вопросы про...

- 1 Неработающий автологин пользователя
- 2 Переименование нескольких файлов
- 3 Как заставить работать компилятор
- 4 Первую попытку использования Linux
- 5 Проблемы при компиляции из исходников
- 6 Резервное копирование файлов с rsync

1 Загадка с паролем

В У меня упорно не получается войти в Ubuntu под своим паролем, в правильности которого я уверен. Изначально я выполнил автоматическую установку, но после этого все равно появилось окошко для ввода логина и пароля, и мой пароль не подошел. Но при загрузке в режим восстановления тот же пароль прекрасно сработал! Затем я зашел в систему как пользователь Guest, и командой `sudo`, которая приняла мой пароль, сбросил пароль своего пользователя `pam1` на исходный, вышел из системы, но по-прежнему не могу в нее войти. Я создал нового пользователя `pam1` с тем же паролем; от его имени войти в систему получается всегда, но я хотел бы решить проблему с первым пользователем, чисто из интереса.

Cushie, с форумов

Сомневаюсь, что проблема в пароле — иначе Вы не смогли бы войти в консоль восстановления; но автологин должен работать. Похоже, что-то препятствует входу в систему через рабочий стол, потому-то окно с логином и паролем и появляется каждый раз, хотя автологин включен: попытка входа в систему не удается, и окно появляется снова. Что касается паролей, то Вы сделали все необходимое, поэтому я почти уверен, что это проблема с правами на запись для файлов или каталогов, с которыми оперирует рабочий стол. Я сталкивался с таким и раньше, когда при попытке решить какую-то проблему владельцем важного каталога или файла становился `root`, и в результате другие пользователи не могли записать в него данные.

К счастью, это легко исправить. Загрузитесь в консоль восстановления и войдите в систему от имени проблемного пользователя, затем выполните команду, которая снова сделает Вас владельцем всех файлов в домашнем каталоге.

```
sudo chown -R ${USER}: ~
```

Команда `chown` позволяет сменить владельца (Change OWNership) файлов. Параметр `-R`

означает, что нужно рекурсивно изменить все файлы по заданному пути. На место переменной `${USER}` будет подставлено Ваше имя пользователя (поэтому и нужно заходить от имени пользователя, с которым есть проблемы). Двоеточие после имени пользователя означает, что также нужно изменить группу владельца файлов на группу по умолчанию этого пользователя. Наконец, вместо `~` оболочка подставляет путь к Вашему домашнему каталогу. Команду нужно запускать с `sudo`, так как для изменения владельца любых файлов и каталогов, которые Вам не принадлежат, нужны права `root`, а это тот самый случай. Если Вы случайно забудете указать `sudo`, `chown` выплетет сообщение об ошибке для каждого файла, права на который не сможет изменить — так мы быстро и легко определим затрагиваемые файлы. Более правильный вариант их поиска — командой `find`:

```
find ~ ! -user ${USER}
```

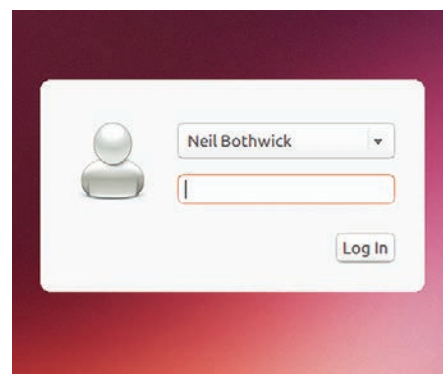
Эта команда означает «найти в домашнем каталоге все файлы и каталоги, владельцем которых не является пользователь `${USER}`». `Chown` не затронет права доступа — для их изменения нужно воспользоваться `chmod`, и изменять их рекурсивно в домашнем каталоге — отнюдь не самая хорошая идея. Некоторым программам нужны особые права доступа к своим файлам, но они не будут иметь ничего против файлов, владельцем которых является запустивший программу пользователь.

2 Осмысленные имена

В Я загрузил с Amazon несколько MP3-файлов с именами наподобие `BOOEXWBVRM_disc_1_09_Why'd_You_Only_Call_Me_When_You're_.mp3`. Не подскажете какой-нибудь быстрый и грубый (или медленный и элегантный) способ очистить имена файлов от мусора, оставив, например, только `Why'd_You_Only_Call_Me_When_You're_.mp3`?

pk_fox с форумов

Это один из тех вопросов, вариантов решения у которых больше, чем файлов, которые нужно переименовать. Можно пропустить имена файлов через `sed` или `tr`, отправив вывод на `mv`. Можно воспользоваться графической программой, например, *Rename*, или одной из утилит командной строки, или командами, предназначенными специально для решения этой задачи, например, `mmv` или `rename`. Команда `rename` входит в пакет `util-linux` и поэтому есть во всех дистрибутивах Linux, но Вам нужна другая программа с точно таким же названием; ее можно



Если вас вернули в окно для ввода логина и пароля, это не обязательно означает, что пароль был набран неверно, особенно если настроен автоматический вход в систему.

загрузить с <http://rename.berlios.de>. В некоторых дистрибутивах она устанавливается под другим именем, например, *renamextm*, а в некоторых исходную *rename* переименовывают в *rename.ul*. То есть разных *rename*, как и разных дистрибутивов, тоже довольно много.

Для выбора файлов в *rename* используются как стандартные, так и расширенные регулярные выражения, в зависимости от указанных параметров. Для Вашего примера команда переименования будет такой:

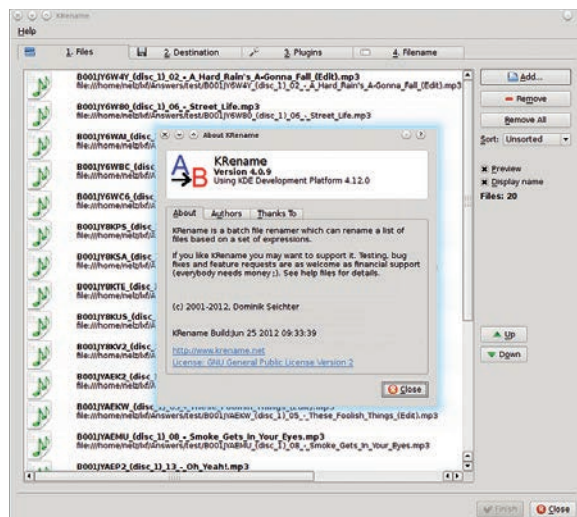
```
rename -t -s/*_[0-9]\[0-9]_ _//r *.mp3
```

Параметр `-t`, или `--test`, означает, что команда будет только выводить результаты на экран, ничего не делая с файлами. Продолжайте запускать команду в таком режиме, пока не достигнете желаемого. Параметр `-s` указывает строки для поиска и замены, разделенные слэшами (точно так же, как `sed`). Обратные слэши необходимы потому, что команда набирается в оболочке, и мы не хотим, чтобы символы `*` и `[]` ею интерпретировались. Строка поиска такова:

```
.*_[0-9][0-9]_ _
```

Это «любое количество символов, за которым следуют `_` и две цифры и `_`», что соответствует Вашему примеру. Строка замены пуста, так что эта часть имени файла просто удаляется, а `r` в конце говорит *Rename* о том, что это простое регулярное выражение. Для расширенных регулярных выражений в стиле *Egrep* используйте `e`. Указывайте критерии возможно подробнее: это снижает риск ошибок.

Наряду с переименованием отдельных файлов и файлов по шаблонам вроде `*.mp3`, *Rename* также может рекурсивно переименовать все файлы по заданному пути и во всех попавшихся подкаталогах. Поскольку *Rename* не трогает файлы,



Хотя вариантов потокового переименования файлов в командной строке великое множество, существует и несколько графических утилит, вроде KRename в KDE.

не соответствующие шаблону, ее можно преспокойно запускать для всего каталога с музыкой каждый раз при добавлении туда новых файлов.

```
rename -R -t -s/\*[0-9]\*[0-9]/*_/_/r ~/Music
```

Когда Вы подберете верные параметры и *Rename* будет выполнять то, что Вам требуется, убедите параметр *-t* или замените го на *-v* и начните переименование.

3 Один верный ПУТЬ?

В Я пытаюсь заставить работать компилятор XDS Modula-2. Каталог *xds* был установлен в */usr/local*, последующие инструкции звучат так: «После установке измените свою переменную окружения *PATH*, добавив в нее путь к каталогу */usr/local/xds/bin*». Я заглянул в разные источники, и в одних говорится, что нужно изменить *PATH* в *.bash_profile*, а другие ссылаются

на */etc/profile*. Какой файл нужно изменить, как именно и какой синтаксис использовать?

Дейв Причард [Dave Pritchard]

Оба местоположения указаны правильно. При загрузке оболочки, еще до появления командной строки, оболочка загружает настройки из файлов профилей. Сначала загружается */etc/profile*, и это глобальный файл, применяемый всеми пользователями оболочки, так что любые изменения в нем окажут влияние на всех пользователей (и на все оболочки — ведь не каждый пользуется *Bash*). А файл *~/.bash_profile* является отдельным для каждого пользователя и находится в его домашнем каталоге. Он загружается после */etc/profile*,

а значит, указанные в нем настройки являются специфичными только для этого пользователя и ими можно перегрузить любые настройки, указанные в */etc/profile*.

Выбор между */etc/profile* и *.bash_profile* зависит от того, сколько у Вас пользователей или один, и от того, всегда ли программа будет запускаться из оболочки *Bash*.

Есть и третий вариант. Во многих дистрибутивах есть каталог */etc/profile.d*. Файлы профилей из этого каталога по очереди загружаются в файле */etc/profile*. Различие только в структуре — вместо того, чтобы редактировать файл */etc/profile* для каждой программы, которой это может понадобиться, у каждой есть свой собственный файл. Конечный результат тот же самый, но многим, включая и меня, такое решение кажется гораздо более элегантным. Такой подход вдобавок

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 запрошен 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 выполняйте любые заданные команды без предшествующей *sudo*.

сохраняет Ваши изменения в профиле от случайной перезаписи после обновления.

Какой бы файл вы ни выбрали, в него нужно добавить следующие строки:

```
PATH="/usr/local/xds/bin:$PATH" export PATH
```

Пути указаны в переменной окружения *PATH* в виде списка путей, разделенных двоеточиями. »



Коротко про...

Screen

Освоившись в командной строке Linux, вы все чаще и чаще будете обращаться к терминалу. *Xterm*, терминал со вкладками, позволяет одновременно работать в нескольких сеансах, но при удаленном подключении к рабочему столу или в виртуальной консоли им особенно не попользуешься. *Screen* — оконный менеджер для терминалов, позволяющий запускать несколько сеансов в одном терминале; но это только начало.

Наберите *screen* в терминале — и хотя с виду ничего, кроме очистки окна, не происходит, вы уже в другом терминале. Наберите *ls -l* для вывода содержимого текущего каталога, затем нажмите *Ctrl+a c*, и оно исчезнет. Перейдите

в другой каталог и снова выполните *ls*, нажмите *Ctrl+a n* — и вы снова видите исходный список. *Ctrl+a c* создает новый терминал, оставляя старый нетронутым; *Ctrl+a n* переключает между ними. Теперь нажмите *Ctrl+a d* для выхода из *Screen* и возврата в исходную оболочку, откройте другой *xterm* и наберите *screen -r*. Вы вернулись в исходный сеанс *Screen*.

Screen не ограничивается локальными подключениями и прекрасно работает по SSH. Например, можно подключиться по SSH к домашнему компьютеру с работы, запустить *Screen*, загрузить новую версию какой-то программы, запустить компиляцию и выйти из *Screen*. Потом, чтобы проверить, как идет работа, можно подключиться к своему сеансу *Screen* командой *screen -r* из офиса или из дома. Даже при разрыве SSH-соединения программы, запущенные в *Screen*, не перестанут выполняться.



В верхней части окна — компиляция программы, в нижней — IRC-чат.

Еще много параметров описано на map-странице. Так, можно разделить монитор комбинацией клавиш *Ctrl+a s*, перейти в новую область с помощью *Ctrl+a Tab* и нажать *Ctrl+a n* (или *c*) для отображения другого терминала *Screen*.

Первая строка добавляет новый путь в список; затем команда **export** делает путь доступным всему окружению (в противном случае изменения будут отменены, когда файл профиля закончит выполняться). Если в файле уже есть команда **PATH** со следующей за ней командой **export**, поместите первую строку между ними — не нужно экспортировать каждое изменение отдельно, так как в конце файла есть общий экспорт.

Чтобы еще больше все запутать — возможно, вы также видели ссылки на файл `~/.profile`. Это эквивалент `/etc/profile` для каждого пользователя, но *Bash* ищет этот файл только в том случае, если `.bash_profile` нет, в противном случае он может игнорироваться. Мы бы посоветовали остановиться на `~/.bash_profile` или `/etc/profile`.

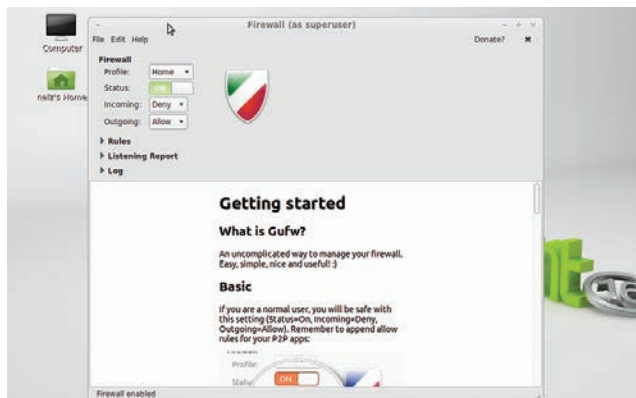
4 С чистого листа

В связи с надвигающимся прекращением поддержки Windows XP я решил попробовать Linux на рабочем, хотя и старом ноутбуке HP Pavilion Notebook zv6115EA. Единственное подключение к Сети, которое у меня есть — мобильный интернет через Mi-Fi модем.

Я приобрел LXF179 и хотел сделать двойную загрузку с Ubuntu 13.10 с диска этого номера, но в итоге загрузил Ubuntu 12.04.3 LTS. Меня очень удивила простота, с которой я смог настроить систему с двойной загрузкой. Но почему установщик отключает беспроводную сетевую карту и не включает ее обратно, и как это исправить? В Windows XP Wi-Fi по-прежнему работает. Я попробовал загрузиться с Live-диска Mint 16, который мне очень нравится, но не смог найти в нем брандмауэр, он там есть? В одном из окон «Идеальной Linux-машины» [LXF180, стр. 48] есть иконка брандмауэра, но при запуске Mint 16 в Live-режиме она отсутствует. Нужно загрузить брандмауэр? Пользуетесь ли вы в Linux антивирусными программами, и если да, то какими? До того, как попробовать ваши диски, я думал, что Linux — система заумная, а теперь понимаю, что это не совсем так.

Уильям Хинд (William Hind)

Вы молодец, что сделали решительный шаг и попробовали что-то новое. Linux, как Вы, без сомнения, убедились, сильно



В Linux брандмауэр не обязателен, но если вы решите установить его, воспользуйтесь *Gufw*.

отличается от Windows, и очень многие вещи требуют другого подхода. Причина, по которой не работает сетевая карта, в том, что драйвер или прошивка для нее не установлены. Не зная модель сетевой карты, мы не можем сказать, какой драйвер нужен, но если Вы откроете терминал и запустите **lspci**, то увидите список устройств на своем компьютере. Найдите в нем свой беспроводной сетевой контроллер — это будет нечто вроде

```
02:00.0 Network controller: Qualcomm Atheros
AR9485 Wireless Network Adapter (rev 01)
```

и поместите его название и модель вместе со словом Linux в строку поиска своей любимой поисковой системы. Так Вы должны узнать, какой драйвер нужен и как его установить. В Windows все работает потому, что в Windows драйвер был установлен вместе с системой.

Иконка брандмауэра, которую Вы видели на экранном снимке, принадлежит *Gufw*, графическому клиенту брандмауэра *Ufw*, установленному в Mint. По умолчанию *Gufw* не установлен, поэтому зайдите в центр управления программами, наберите **gufw** в строке поиска и установите его. В Linux брандмауэры используются не так, как в Windows. Сами функции брандмауэра в Linux предоставляются ядром, а внешние программы лишь помогают настроить их использование.

Одно из основных назначений брандмауэров в Windows — блокировать исходящие соединения от нежелательных программ. С открытым ПО этой проблемы нет, так как скрывать вредоносный код в программах, код которых при желании может просмотреть любой, малопрактично. Если Вы будете устанавливать программы только через менеджер пакетов или центр программ, то брандмауэр на исходящие соединения Вам не нужен. В дистрибутивах также обычно по умолчанию не устанавливаются программы, принимающие входящие соединения, поэтому закрывать их брандмауэром не нужно, если только у Вас не запущены интернет-сервисы. Впрочем, лишним брандмауэр не будет.

Вирусов для Linux практически не существует, поэтому большинство пользователей считают антивирус ненужным. Если Вы все равно хотите поставить какой-то антивирус, лучший вариант — *ClamAV* из менеджера пакетов дистрибутива. Несмотря на то, что его главное назначение —

сканирование на вирусы Windows файлов, которыми Вы намерены делиться с пользователями Windows, регулярный запуск антивируса даст Вам ощущение спокойствия.

5 Повтор копирования

В Я делаю резервную копию своего сервера на локальный USB-диск и сетевые диски с помощью *rsync*. Все работает отлично, за исключением файлов с расширением

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, поскольку проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам нужно знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства.

Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (<http://hardinfo.berlios.de/>) — она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файле, который вы сможете приложить к своему письму.

Не уступающий в удобстве альтернативный вариант — *Ishw* (<http://ezix.org/project/wiki/HardwareLiSter>). Одна из указанных программ непременно должна быть включена в ваш дистрибутив (а иногда и обе).

Если вы не хотите или не можете их установить, выполните следующие команды в терминале от имени root и приложите файл **system.txt** к письму. Это здорово поможет диагностике.

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

.wtv, которые записываются с моего центра мультимедиа в гостиной и копируются снова при каждом резервном копировании. Это отнимает очень много времени. Команда резервного копирования такова:

```
rsync -vare ssh --progress --omit-dir-times --delete --exclude .Trash* --exclude lost+found* /media/Drive-F1/ /media/Drive-NV
```

Рэг Коуэлл (Reg Cowell)

В параметрах *rsync* нет ничего, означающего, что эти файлы будут обрабатываться иначе, чем все остальные; значит, они все-таки изменяются. Так как их использует медиа-центр, возможно, программа что-то с ними делает в ходе своей работы. Она может модифицировать их содержимое или хотя бы временные метки, например, в результате индексации.

Первый шаг — определить, как изменяются файлы. Прежде чем запустить резервное копирование, взгляните на исходный файл и файл в резервной копии и проверьте, не изменились ли дата и размер файла. Если размер не изменился, проверьте, не изменилось ли содержимое, с помощью программы для проверки контрольной суммы, такой как *md5sum*. Если изменяется только временная метка, а размер не изменяется, пропустить эти файлы можно, указав параметр **--size-only** для *rsync*. Но он может затронуть другие файлы, поэтому безопаснее будет запустить *rsync* дважды: первый раз — с параметром **--exclude *.wtv**, а второй — с параметрами **--size-only** и **--include *.wtv**. Например, так:

```
rsync -av --exclude .Trash* --exclude lost+found*
--exclude *.wtv SRC DEST
```

Если содержимое файла тоже немного изменилось, как видно из размера или контрольной суммы, можно вместо `--size-only` воспользоваться параметром `--ignore-existing`, но это следует делать только в отдельном запуске `rsync` для файлов `WTV`. Кстати, несколько из указанных вами параметров не обязательны. Например, для локального резервного копирования `-e ssh` игнорируется, а для удаленного он включен по умолчанию. Также параметр `-r` заключен в параметре `-a`, и его не нужно указывать отдельно.

Есть и другой вариант резервного копирования, который потребует чуть больше работы, но в результате Вы получите хорошую систему резервного копирования. Если на компьютере назначения запустить демон `rsync`, он будет вычислять различия между различными версиями одного и того же файла и передавать только измененные блоки, а не весь файл целиком.

Я делаю это при загрузке ISO-образов для DVD Linux Format, и измененный образ загружается всего за несколько минут, а не за несколько часов. Каталог назначения нужно указать в файле `/etc/rsyncd.conf` в целевой системе:

```
path = /media/Drive-N
comment = Backup directory
```

Затем запустите резервное копирование, указав имя хоста целевой системы (или "localhost" для USB-диска) и имя места назначения `rsync`, заданное в `rsyncd.conf`:

```
rsync -va --progress --omit-dir-times --delete
--exclude .Trash* --exclude lost+found* / media/
Drive-F1/ rsync://hostname/backup/
```

Это позволит копировать и большие файлы, не тратя много времени на передачу файлов.

6 Сборка из исходников

В Я пробую компилировать программы из исходного кода и пока скомпилировал две. Не могу скомпилировать следующие три, так как выдается сообщение об отсутствии файла `.configure`. Программы, которые я пытаюсь скомпилировать — *Nightingale*, *FlightGear* и *Kompozer*. Я могу загрузить их в виде Deb-пакетов, но хочу скомпилировать их. Пробовал *Alien*, но она создает файлы Deb с ошибками.

Narendra

Процесс сборки `./configure && make && sudo make install` свойственен программам, использующим GNU *Autotools*. Это

подавляющее большинство открытых программ, но ни в коем случае не все. Обязательно поищите инструкции по установке в корневом каталоге с исходным кодом. Обычно, хотя и не всегда, они находятся в файле **Readme** или **Install**. Даже если программа использует *Autotools*, нужно взглянуть в инструкции по установке на предмет каких-то особых шагов. Если инструкций по установке нет, взгляните на сайт или в вики проекта.

В случае с *FlightGear* в архиве есть файл **install**, где написано, что для сборки используется **cmake**. Это более новый подход, который набирает популярность, особенно в кроссплатформенных проектах. В этом файле перечисляется, какие действия нужно выполнить для сборки.

Хорошо, что Вы пытаетесь компилировать программы сами, а не полагаться на готовые пакеты — просто сначала прочтите инструкции, и *Alien* Вам здесь не поможет. Эта программа преобразует двоичные форматы файлов друг в друга, что бывает удобно, если Вы хотите установить пакет RPM в системе на базе Debian. Вам также может пригодиться *Checkinstall* в каталоге **Essentials** на всех дисках **LXF DVD**. Эта программа заменяет фазу установки при сборке на сборку пакета в выбранном формате, который затем можно установить с помощью менеджера пакетов. **LXF**



Часто задаваемые вопросы

WPA

» Что такое WPA?

WPA (Wi-Fi Protected Access — защищенный доступ по Wi-Fi) — система, которая гарантирует приватность данных, передаваемых по беспроводным сетям.

» А оно мне надо?

Если вы живете не в экранированном бункере, то почти наверняка да. В отличие от традиционной проводной сети, в которой ваши данные никогда не покидают проводов, соединяющих компьютеры, беспроводная сеть распространяет их на сотни метров. Это значит, что любой, у кого есть ноутбук и сетевая карта для беспроводной сети, может так же легко прочесть ваши данные, как если бы он подключился к маршрутизатору вашей сети.

» У меня большой дом, ноутбук даже не подключен к сети в остальных комнатах, поэтому я в безопасности, разве нет?

Нет! Хотя начиная с определенного расстояния, ваш ноутбук может не подключиться к точке доступа,

зато другие устройства смогут. Прикрепив к беспроводной сетевой карте антенну, вы получите гораздо большую дальность. В этом может не быть ничего страшного, если вы пользуетесь Wi-Fi только дома для выхода в Интернет с ноутбука. Ну, а если ваша компания пользуется сетью Wi-Fi со слабой защитой (или без нее), а кто-то в припаркованном снаружи автомобиле считывает все конфиденциальные данные?

» Раз уж об этом зашла речь, что такое WEP?

WEP — Wired Equivalent Privacy (приватность, эквивалентная проводной сети). Это была предыдущая попытка защиты беспроводных сетей. Хотя это лучше, чем совсем не иметь защиты, взломать ее не так трудно. Она может спасти вас от соседей, которые случайно подключатся к вашей точке доступа вместо своей собственной, но не спасет от тех, у кого есть намерение взломать ваше соединение.

» Какие устройства необходимы для WPA?

Все новые беспроводные сетевые карты и точки доступа поддерживают WPA и WEP. Если у вас более старое устройство, может оказаться, что оно поддерживает только WEP.

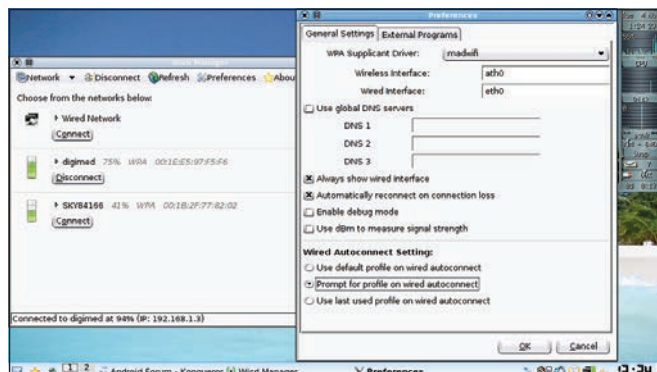
» Как насчет ПО?

Вам нужна поддержка WPA на двух уровнях — на уровне драйверов устройства и в отдельной программе, реализующей WPA. Самая популярная из таких программ — *wpa_supplicant*, она должна быть в вашем дистрибутиве. Если ее там нет, загрузить ее можно с сайта http://hostap.epitest.fi/wpa_supplicant.

Эта программа поддерживает аутентификацию и обмен ключами.

» Наверное, чтобы воспользоваться WPA в Linux, придется читать ман-страницы и редактировать файлы настройки?

Можете так и сделать, если жизнь в *Vim* прибавляет вам ярких эмоций. Но, к счастью, есть и графические утилиты, которые могут упростить жизнь. Самая продвинутая из них, пожалуй, *Network Manager* в Gnome (<http://bit.ly/1edR8hS>). Хотя проект сделан для Gnome, в нем есть клиенты и для других утилит.



» Network Manager — не единственный вариант: wicd также прекрасно управляет проводными и беспроводными соединениями.



LXF HotPicks



Ричард Смедли

Активист FOSS, Ричард колесит по грязным полям Interweb на своем верном SMED5000 в поисках новых залежей сокровищ открытого кода для вас.

Enlightenment E18 » GnuCash » DevCoin » abc2midi » GNU Octave » Shell AudioScope » Brackets » PlaneShift » Hnefatafl » MKVToolNix » GNU Grep

Среда рабочего стола и менеджер окон

Enlightenment

Версия: E18 Сайт: www.enlightenment.org

E17 был в разработке 12 лет. Вызревание *E18*, которое длилось всего лишь год, по сравнению с этим кажется поспешным, однако 242 изменения намекают на серьезное улучшение. *Enlightenment* достаточно просто устанавливается на Ubuntu Saucy (кому-нибудь еще слышится, как Кеннет Вильямс [Kenneth Williams, актер-комик] произносит: "O, saucy"? Только мне? Ну, ладно) с включенным репозиторием Universe и добавленным еще одного PPA:

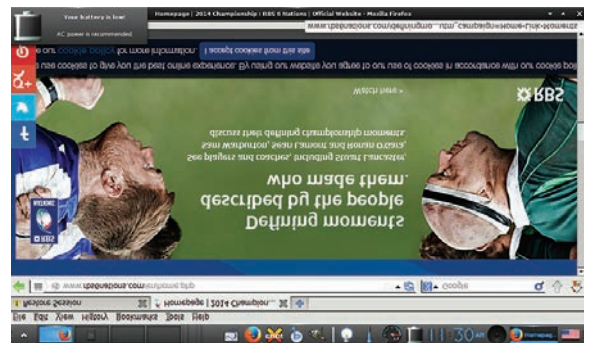
```
sudo add-apt-repository ppa:vase/ppa
sudo apt-get update
sudo apt-get install enlightenment
```

Когда вы будете читать эти HotPicks, пакеты для других дистрибутивов уже должны найтись.

Э, а почему картинка вверх ногами? Наше тестовое оборудование Ubuntu пало жертвой ошибки *EGL* плюс mesa-драйвера *GLS*. Если вы тоже пострадали от нее, то рекомпиляция с *glx/desktop-glx* спасет ситуацию, но к моменту чтения вами этого материала все уже могут исправить.

Загрузка Ubuntu PPA на наш компьютер Debian Sid дало нам возможность простой установки, и неприятностей больше не было. *Enlightenment* отличается высочайшей степенью индивидуальной настройки —

«E18 готов к прекрасному новому миру конвергенции.»



» Упс! В *EGL/GLES* есть ошибка драйвера Intel mesa, которая ставит содержание с ног на голову, если не сделать рекомпиляцию с помощью *glx*.

и не через текстовые файлы, а через диалоговые окна. Войдя в новую среду, вы выбираете язык, раскладку клавиатуры и то, будете ли вы использовать мобильное устройство с сенсорным экраном или стандартный ПК с клавиатурным интерфейсом. Да, подобно Ubuntu 14.04, E18 готов к прекрасному новому миру конвергенции мобильных устройств и ПК.

Согласившись с настройками по умолчанию для размера декора окон, модели фокуса и т.д., мы получим свой идеал рабочего стола, и все настройки можно изменить позднее. Первое впечатление — в частности, на старом оборудовании — что рабочий стол очень отзывчив; затем видишь, насколько он привлекателен внешне и насколько он создает ощущение NeXT-Step. А все благодаря библиотекам *Enlightenment Foundation Libraries (EFL)*.

Разработанные за долгие годы для *E17*, *EFL* включают: *Evas*, библиотеку «холстов [canvas]»; *Edje*, движок раскладки; набор виджетов *Elementary*; плюс множество других вспомогательных библиотек, которые разделяют графические и функциональные части программ, значительно облегчая работу кодеров и дизайнеров.

Далее, *E18* улучшил менеджер файлов и предлагает лучшую интеграцию рабочего стола для всего, от Bluetooth до музыки. Да еще композитинг встроен в ядро и повышена стабильность. *E18* должен обрести много новых друзей.

Исследуем интерфейс Enlightenment 18

Прозрачные окна

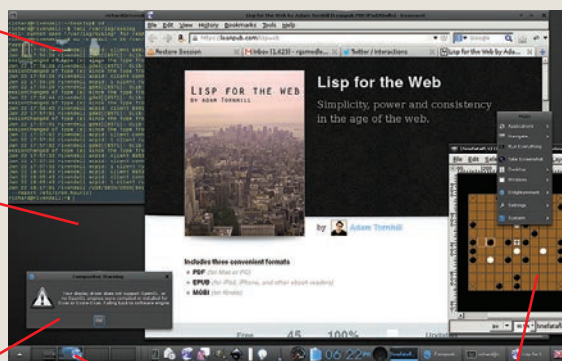
Прозрачность — функция не новая, но Terminology добавляет в нее некий шик.

Настройка цвета

Темные цвета возвращаются, но если вам нужен более солнечный рабочий стол — *E18* легко поддается настройке.

Разгон оборудования

Enlightenment работает с любым вашим ускорением оборудования, но достаточно нетребователен, чтобы поладить и со старым оборудованием.



Панель задач

Enlightenment предлагает знаковую панель задач для переключения между меню, рабочими столами и окнами; для виджетов и значков быстрого запуска.

Настройка через GUI

В отличие от большинства приложений Unix, *E18* не просит редактировать текстовые файлы настройки: каждая настройка делается за один-два щелчка.

Бухучет для малого предприятия

GnuCash

Версия: 2.6 Сайт: <http://gnucash.org>

Удобство — и стоимость — стимулируют переход на «облако» малых предприятий и частных лиц, однако цена этого удобства — конфиденциальность. Так что для финансовой информации, вероятно, есть смысл настроить собственную систему и использовать отдельную программу.

Gnucash уже 15 лет предлагает свободную бухгалтерскую программу с открытым кодом для пользователей Linux, постоянно добавляя функции для предприятий и частных лиц, а также улучшения движка, такие, как чистые арифметические операции с фиксированной запятой, защищающие пользователей от погрешностей округления. В нем есть свои хитрости (например, все является «счетом [account]»), включая категории расходов, однако он очень солиден и надежен благодаря: двойной записи; поддержке фондов в разной валюте; счетам кредиторов и счетам дебиторов; и почти все, что он предлагает, он делает очень

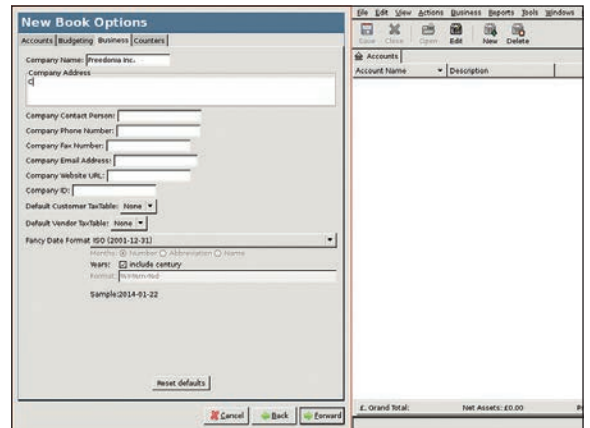
хорошо, кроме, вероятно, его функций бюджета.

В вашем дистрибутиве есть пакет для него; установка несложная, однако всем, кто не пользуется Gnome, надо быть готовыми к установке других пакетов. Во внешнем виде этой программы нет ничего сверхсовременного, что, как нам кажется, вполне нормально для финансового пакета. Тем не менее в версии 2.6 появились визуальные улучшения; *jqplot* используется для рендеринга пакетов новых отчетов, и обновлена система управления преднастроенными отчетами.

Выгоды бизнеса

Для бизнес-пользователей мы увидели следующие добавления: кредитное авизо;

«Солидные и надежные функции бухучета с двойной записью.»



» Большая часть настройки делается через мастеров. Если у вас есть счета в Quicken или MS Money, можете их импортировать.

общие сведения о клиенте и производителе; улучшенная работа со счетами (выставление, печать или дублирование нескольких пунктов прямо из списка результатов поиска). А также способность перераспределять существующие транзакции в качестве платежей по счетам.

Остальные улучшения в 2.6 — ряд мелочей, призванных значительно облегчить жизнь пользователя: например, прямой экспорт PDF; выделение цветом любого счета; открытие заблокированных книг только на чтение; сохранение по фильтру даты. *Gnucash* не столь блистателен, как *KMyMoney*, но бизнес-пользователям его можно рекомендовать не колеблясь.

Криптовалюта

Devcoin

Версия: 1.0.12 Сайт: <http://devcoin.org>

Повальное увлечение криптовалютой пока что не проходит, и мы наблюдаем еще и ряд интересных побочных эффектов, таких, как большой общественный интерес к природе иллюзии, которую мы договорились называть деньгами. Те же и Devcoin: «проект на базе криптовалюты Bitcoin, вдохновленный принципами этики и созданный для содействия финансирования открытых проектов писателей, художников и разработчиков ПО».

С технической точки зрения, это очень близко к Bitcoin. Фактически, они добываются совместно. Добывающие биткойны могут одновременно добывать и девкойны: 50000 монет за блок, и 90% идут на финансирование проектов Открытого кода, 5000 отдаются добытчику, а 45000 распределяются между художниками и участниками проектов.

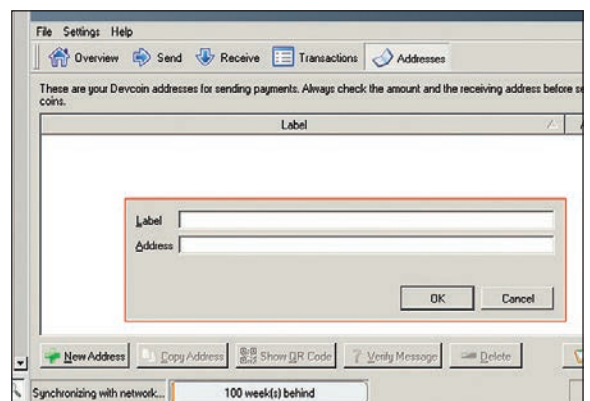
После скачивания и установки файла *.jar* для Linux вы запускаете кошелек.

Он должен синхронизироваться с огромным количеством данных, и, в отличие от Bitcoin, здесь нет лимита на произведенные Devcoins, хотя количество Devcoins, выпускаемых ежегодно, фиксировано (2,6 млрд); таким образом, с течением времени инфляционные факторы должны снизиться. Произойдет ли это на практике — вопрос спорный, учитывая, что обмен криптовалютой на национальные валюты оброс всякими домыслами.

Что у вас в кошельке?

Следующий шаг — заработать монеты. На Devtome (<http://bit.ly/1nclUYy>) есть список премий за написание ПО с открытым кодом. Он включает создание родного

«Проект на базе Bitcoin, вдохновленный принципами этики.»



» Прежде чем начинать перечислять и получать платежи, надо синхронизировать 100 недель торговых транзакций...

кошелька Devcoin для Android, вместе с кошельком Goldcoin (<http://gldcoin.com>); биржу и рынок Devcoin для обмена биткойнов на девкойны; а также зарабатывание девкойнов посредством написания статей на Devtome wiki; и даже реклама.

В настоящее время проект, за который начисляется премия, выбирается посредством обсуждения (очень длинная ветка) на <http://bitcointalk.org>. Возможно, для тех, кто настроен скептически, этого недостаточно, но если вы все равно добываете биткойны, то это простой способ поддержать FOSS.

Текстовая запись для пакета конвертации MIDI

Abc2midi

Версия: 20140112 Сайт: <http://abcplus.sf.net>

В прошлом месяце мы нахваливали *Philip's Music Writer (PMW)* за создание прекрасных партитур. *Abc* — основанная на тексте нотная запись, которой легко пользоваться, и она приобрела способность распечатывать весьма сложные партитуры с помощью утилит вроде *abc2mps*.

Однако настоящий рывок для форматов — это *abc2midi*, который, как можно догадаться, производит MIDI-файлы из текстовых файлов в нотации *Abc*. Добавьте 400000 мелодий с <http://abcnotation.com> и совместные возможности разнообразных *wiki Abc*, и вот вам формат и набор инструментов, который стоит попробовать любому меломану.

В вашем дистрибутиве может быть версия *abcmidi* и связанных с ней инструментов пятилетней давности (или даже больше), но это не может служить отговоркой — на сайте есть пакеты Deb и RPM релиза этого месяца всего пакета *abcMIDI (abc2midi)*, плюс инструменты проверки форматов, переформатирования

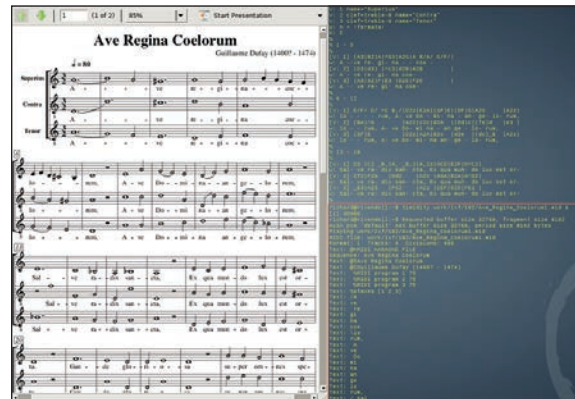
и транспозиции), а также утилиты для редактирования *Abc*, конвертирования в *LaTeX* и извлечения отдельных частей.

Расширения синтаксиса для *Abc* позволяют *abc2midi* справляться с довольно сложным переложением отдельных частей и управлением длительностью отдельных нот, а слова могут отображаться в файлах Karaoke MIDI.

Вносим гармонию

На практике *abc2midi* обработала ряд хороших и народных мелодий, которые мы нашли на разных сайтах *Abc*. Там накопилась настоящая музыкальная сокровищница, и вы можете легко добавить собственную мелодию. Нотация начинается со временной подписи (M:) и длительности нот по умолчанию (L:), и т.д.:

«Набор инструментов; стоит попробовать любому меломану.»



Музыкальный файл *Abc* (вверху справа), конвертированный в MIDI (внизу справа) и Postscript (слева) — все из текстовой записи.

```
X:11
T:God Save the King
A:Wyresdale, Lancashire
B:A John Winder Book, April 1789
L:1/8
M:3/4
Затем ноты, отделенные | (значок такта):
d2d2 e2 | c3d e2 | f2f2 g2 | f3e d2 | e2d2
c2 | d6 ::
a2a2 a2 | a3g f2 | g2g2 g2 | g3f e2 | \
f2 (gf)(ed) | f3g a2 | ((3bag) f2e2 | d6:|
```

Есть определенные компромиссы между простым вводом *Abc* и сложным результатом созданной партитуры, которые повлияют на то, выберете ли вы ее или *PMW*, *LilyPond* или *MusicTeX*, однако если рассматривать его как генератор MIDI, то *Abc* — отличная штука.

Вычислительный язык программирования

Octave

Версия: 3.8.0 Сайт: www.octave.org

GNU Octave — высокоуровневый язык для числовых вычислений, линейных и нелинейных, в основном совместимый с Matlab. Некогда просто приложение командной строки, сейчас он имеет опциональный GUI, который станет интерфейсом по умолчанию в 4.0.

Для большинства задач, выполняемых в GNU Octave, можно с тем же успехом использовать R (или GNU S), или Python и его превосходные библиотеки (включая *numpy* и даже *matplotlib*), однако колледжи стремятся придерживаться стандарта Matlab или Octave. Octave появился не в качестве клона, но современной разработкой по большей части и движет потребность в совместимости с Matlab.

Если вы компилируете исходник, вам нужен компилятор Fortran77 и библиотеки *OpenBLAS* и *Lapack*, а также *glib2*, если вы хотите распечатать или сохранить

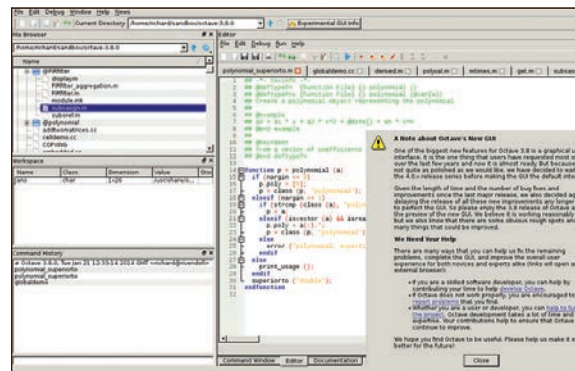
цифры с графикой OpenGL. Компиляция весьма долгая, так что можете отправиться за чайником. В качестве альтернативы в дистрибутивах начали появляться пакеты, и когда **LXF** выйдет в продажу, скорее всего, в тестовом репозитории вашего дистрибутива уже появится GNU Octave 3.8.

Желая крутануть новый GUI, начните с `octave --force-gui &` и первое, что вы заметите — для создания GUI разработчики выбрали *FLTK* и *OpenGL*.

Дружба с Matlab

Здесь множество новых функций, от плottера до вложенных функций с правилами видимости согласно Matlab. Есть

«Разработкой движет потребность в совместимости с Matlab.»



Наличие опционального GUI — смелый новый шаг для GNU Octave. Начиная со следующего релиза, это будет интерфейс пользователя по умолчанию.

парсер TeX, ограниченная поддержка именovaných исключений и множество новых функций и настроек для совместимости с Matlab. *GNU Plot* по-прежнему доступен тем, кто не может работать с новым плottером OpenGL. Несколько пакетов расширений (наборов инструментов) имеются на <http://octave.sf.net>.

Взяв код Matlab из колледжа, вы обнаружите, что по большей части он работает, с минимумом модификаций, если только не зависит от конкретных наборов инструментов. Может потребоваться чистка нестабильного синтаксиса, но добавив опции в `~/.octaverc`, вы получите предупреждения, если ваш код будет несовместим.

Аудиоосциллограф в текстовом режиме

AudioScope

Версия: 0.21.80 Сайт: <http://bit.ly/1eMWaDQ>

АudioScope для оболочки — это аудиоосциллограф в текстовом режиме, который фиксирует звуки из `/dev/dsp`, `SoX` и демо-версий, предоставляя вам всю красоту графики ASCII в оболочке `Bash`.

Только когда я скачал скрипт, прочитал исходник и онлайн-материал по ссылке, я обнаружил, что автор, Барри Уокер [Barry Walker], разработал это приложение частично в результате обсуждений на форумах **LXF** (<http://bit.ly/1b150gn>). Фактически, месяцы обсуждений на форумах — это уже само по себе история *AudioScope.sh*!

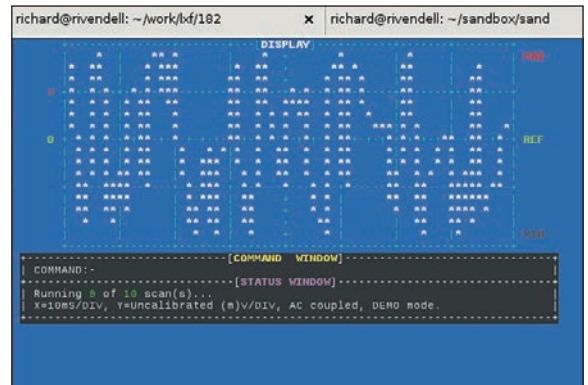
Написанный как упражнение для изучения скриптов `Bash`, он быстро превратился в полезный инструмент обучения для детей: скрипт и Readme содержат монтажную схему и инструкции по созданию эталонного устройства. Рисунки схем для калибровки и генерирования волн можно также найти в ветви форума. Суть в том, что все это в пределах возможностей 10-летнего ребенка при умеренном

руководстве. Скрипт включает код и монтажную схему для калибровки горизонтальных и вертикальных осей. На этот раз схем мы не строили — но нам они определенно понравились, и мы бы хотели видеть больше возможностей модификации оборудования, проникающих в программы Linux.

Упражнение в Bash

AudioScope.sh предлагает демо-режим без всяких зависимостей, который берет за образец `/dev/urandom/`. При первом запуске он оказывается в демо-режиме, с одноразовой разверткой и неоткалиброванной самой высокой скоростью системы отсчета времени. Опции можно найти через Readme (самая последняя версия — там же, где и скрипт) или исходный код

«Частично разработан в ветке обсуждения на форумах LXF.»



» Вся красота звука в достоянии ASCII.

для переключения изображения в режим Linux `/dev/dsp` или `SOX`; изменения скорости системы отсчета времени; вызова подсказки; автосохранения сканированных изображений и других изменений в параметрах.

Readme еще указывает на ветку на <http://unix.com>, где тоже показана разработка. Любому, кто заинтересован в изучении полезных скриптов BASH, рекомендуется взглянуть на подобный развивающийся проект, чтобы увидеть как возмозности скромной оболочки, так и развитие проекта. Когда мы сдавались в печать, скрипт только что появился в репозитории GitHub, и теперь вы можете взять его с <http://bit.ly/1bR6MPj>.

Редактор HTML

Brackets

Версия: Sprint 35 build Сайт: <http://brackets.io>

Перед вами — свободное приложение от Adobe. Минуточку... от Adobe? Да! А теперь и в Linux-версии. Brackets — это редактор HTML, созданный на web-технологиях — иными словами, на JavaScript, CSS и HTML. Однако в данный момент он работает внутри маленького родного приложения `C++/GTK, Brackets-shell`.

Уровень заинтересованности в участии сообщества впечатляет — приоритетом для команды являются запросы извне Adobe; поощряются загрузки расширений реестра *Brackets* и даже ведутся списки таких расширений, которые не были загружены. Вклад сообщества особо подчеркивается в примечаниях к релизу.

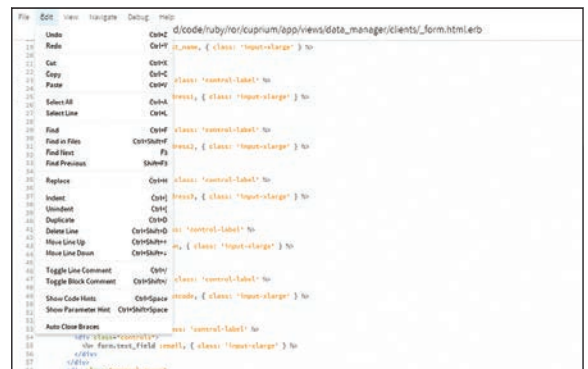
Linux-версия появилась довольно поздно, однако быстро нагнала релизы Mac и Windows. Самая последняя версия, Sprint 35, предлагает улучшения

в интерфейсе Find and Replace, и на 50% уменьшенное время запуска (благодаря минимизации).

Фишка *Brackets* — предпросмотр в режиме live, пока только в браузере *Chrome*. Когда вы настроите CSS сайта, браузер немедленно обновится для показа изменений, и выбор свойства в Brackets выделяет его в браузере.

Маленький и чуткий

Theseus — это отладчик JavaScript в *Brackets*, для *Chrome* и приложений Node.js, который предоставляет проверку событий и обновляет счет вызовов, например, когда на странице нажимается кнопка. Его



» *Brackets* предлагает только опции, нужные в понятном и незахламленном редакторе HTML.

даже можно использовать, чтобы раскурочить сам *Brackets*.

Brackets очень хорош для программ на стадии альфа, и в нем множество расширений: от Ruby и расширения *PhoneGap* до интеграции Dropbox и контекстного меню для вставки спецсимволов HTML. Подключаемая файловая система означает, что Dropbox не является единственной опцией, а на стадии разработки находится более тесная интеграция с GitHub.

Документация разработчиков превосходная, а документация пользователя довольно неплоха для такой быстродвижущейся цели. Впереди немалая работа, поскольку *Brackets* движется к релизу 1.0.

«Его фишка — предпросмотр в режиме live, пока только в Chrome.»

HotGames Развлекательные приложения

Многопользовательская ролевка онлайн

PlaneShift

Версия: 0.6.0 Сайт: www.planeshift.it

Эта MMORPG существует уже довольно долго, даже в виде повторного выхода, и в каждом важном релизе имеются важные улучшения. Разработку начали Atomic Blue, и они по-прежнему ею занимаются, но большая ее часть в руках сообщества, и хотя игра еще не завершена, в ней есть цветущий и сложный мир, который сумеет вас захватить.

Те, кому не повезло с пропускной способностью, должны будут дождаться файла установки объемом 1 ГБ, который придется скачать. Для локальной установки он запускается от имени обычного пользователя, включая возможную установку на USB-брелок, если у вас компьютер с небольшим SSD. Для соединения с игровым сервером требуется регистрация. Перейдите на www.planeshift.it/account/create, а не на неверный URL

с экрана входа в игру. Перед созданием персонажа мы бы предложили сначала немного почитать, поскольку там есть неплохой выбор гуманоидных рас.

Некоторым нравятся ролевые игры, особенно миры фэнтези, но чтобы привлечь тех, кто не в большом восторге от этого жанра, игра обязана быть проста. В этом отношении с *Planeshift* все хорошо, учитывая количество обучающих ресурсов для альфа-релиза. В ней есть встроенная помощь, канал чата, подробная документация на сайте и руководство, которое можно скачать, а также оживленный форум, и все это

«Детализированный интерактивный мир, с массой открытий.»



» *Planeshift* имеет сильное сообщество, и его советы и подсказки помогут вам сделать первые шаги в этом мире.

вместе взятое демонстрирует поддержку сообщества.

Это очень детализированный интерактивный мир, с массой открытий и толпой неигровых персонажей (NPC), которые предложат вам квесты и будут отвлекать от основной линии, а это позволит вам пойти собственным путем без навязанной сюжетной линии, характерной для многих MMORPG.

Древнескандинавская настольная игра

Hnefatafl

Версия: 140117 Сайт: <http://hnefatafl.se>

Викинги наступают! Мало того, что они заполнили телеэкраны своими интригующими преступлениями и парламентскими драмами — теперь они пытаются увлечь нас игрой, упоминающейся аж в древнескандинавских сагах. Появившаяся в Швеции, *Hnefatafl* — это игра двух неравных сторон, согласно более информированному источнику на <http://boardgamegeek.com>: «Защищающаяся сторона имеет двенадцать солдат и короля, который начинает игру в кресте посреди поля. Целью игры для короля является спастись бегством в один из четырех угловых квадратов».

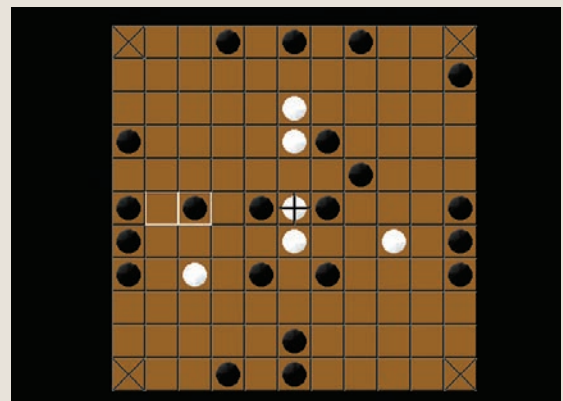
Играть можно на доске 11×11 или 13×13, и если у вас, скажем, есть доска Го, можете попробовать запустить 3D-принтер и сделать несколько игроков. А пока вы возитесь с отладкой своего принтера, запустите скачивание этой

на глазах хорошеющей версии для Linux от Александра Сёдерлунда [Alexander Söderlund] и прогоните шведскую страницу через Google Translate.

Мы установили пакет *libxpm-dev* для файла заголовка *xpm*, изменили язык по умолчанию со шведского [swe] на английский [eng] в *config.mk*, затем запустили *make*; *make install* и были счастливы увидеть *Hnefatafl* в меню. Включены и другие игры *tafl* — см. страницу (на английском!), где вы увидите опции, а также подробности правил и ходов.

Оставив *Hnefatafl* 11×11 по умолчанию, мы ввели P 1 M 1, чтобы игрок 1

«Идеальна для моментов, когда надо переключить себе мозги.»



» Да, компьютер нас победил, но, изучив правила, вы увидите, что сможете его обыграть.

(черные) был машиной 1 (AI), а мы играли белыми.

Будучи новичками, мы продули довольно быстро, но посещение <http://tafl.cynningstan.org.uk> для ознакомления с правилами и стратегией помогло нам превзойти нынешний уровень AI. Игра проходит быстро, и это делает ее идеальной для тех моментов, когда надо за пару минут переключить себе мозги, чтобы снова вернуться к работе.

Утилиты видео- и аудиофайлов

MKVToolNix

Версия: 6.7.0 Сайт: <http://bit.ly/1dEFet5>

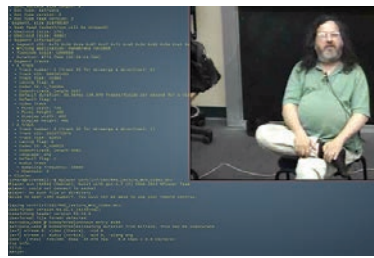
Хотя Ogg — отличный свободный контейнер для тех форматов, которые он поддерживает, Matroska — названный в честь русской матрешки — это контейнер с открытым форматом, который поддерживает практически все форматы аудио: от WMA и DTS до ALAC и MLP/Dolby TrueHD. Еще он поддерживает все видеокодеки, и, похоже, больше форматов субтитров, чем любой другой, а WebM в HTML5 — подмножество контейнера Matroska.

MKVToolNix — это набор инструментов для создания, изменения и проверки файлов Matroska (включая WebM). После более чем десяти лет разработки они вполне устойчивы, но продолжают улучшаться и получать обновления функций с каждым основным релизом, которые выходят регулярно.

MKVToolNix содержит *mkvmerge* для создания файлов Matroska из других форматов. Он использует *mkvinfo* для получения информации о треках в файлах

Matroska и *mkvextract*, чтобы извлечь треки из файлов Matroska в другие форматы. Первые два инструмента также доступны в версиях GUI, что может быть удобно для некоторых операций, например, для перетаскивания частей в видео.

Вы также можете воспользоваться *mkvmerge*, чтобы стянуть в Matroska файл из другого контейнера, например, AVI или MP4, но если вы кодируете с нуля, обратите внимание, что *FFmpeg* тоже поддерживает этот формат. Что касается аудио, очень легко перевести слова в формат SRT и добавить их в файл MKA.



► **RMS** — в свободном видеокодеке и свободном формате контейнера, естественно.

Утилита поиска по тексту

GNU Grep

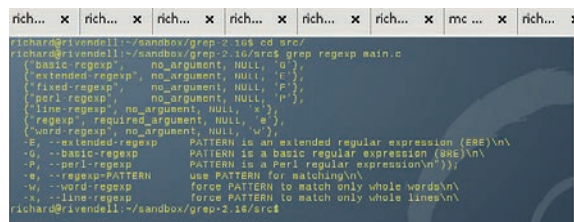
Версия: 2.16 Сайт: www.gnu.org/software/grep

Grep так часто используется в повседневной жизни пользователя в оболочке, что легко забыть, что он не встроен в *Bash*. Эта почтенная утилита происходит от парсера регулярных выражений Кена Томпсона [Ken Thompson] для *ed*, ныне — отдельного приложения, имеющегося в каждом дистрибутиве Unix; *grep* уже лет десять присутствует в Оксфордском словаре английского, в виде как существительного, так и глагола.

GNU-версия *grep* (она включает *egrep* и *fgrep*) позиционируется как «самый быстрый на западе *grep*» благодаря своему детерминированному обнаружителю совпадений с отложенным состоянием и применению поиска Boyer–Moore–Gosper для фиксированной строковой переменной, без просмотра каждого символа. Релизы последней пары лет произвели улучшение ряда мелочей, на которые были жалобы — например, медленный рекурсивный поиск в громоздких иерархиях

директорий; а в последнем релизе решены проблемы многобайтовости и сильно увеличена скорость *grep -P*, всего лишь с помощью своевременной компиляции. *-P* — это опция регулярного выражения Perl.

Кому нужен текстовый поиск с загадочными символами, советуем прочесть *Perl One-Liners: 130 Programs that Get Things Done* Петериса Круминьша [Peteris Krumins]; повысить эффективность вашей жизни в Unix также поможет колонка Сандры Хенри-Стокер [Sandra Henry-Stocker], *Unix as a Second Language on Resolutions* (<http://bit.ly/1jsPzk4>). **LXF**



► Работа в *grep* с исходником *grep* для обработки регулярных выражений — это не слишком мета-операция?

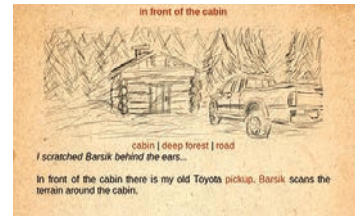
Также вышли

Новые и обновленные программы, тоже достойные внимания...

► INSTEAD 1.9.1.1

Интерпретатор простых текстовых приложений, смесь визуальных новшеств и текстовых квестов.

<http://instead.sourceforge.net>



► Вернитесь в мир простой графики и захватывающих сюжетов.

► 240p-test-suite 1.18

Тестовый пакет для результатов игровых консолей.

<http://bit.ly/1bREvbx>

► OpenPetra 0.2.22

Отличная программа администрирования для некоммерческих организаций.

<http://www.openpetra.org>

► jfuzzylite 1.0

Библиотека Java для управления интеллектуальной логикой.

<http://fuzzylite.com>

► Csound 6.02

Синтезатор звуков и музыки для сочинения и исполнения.

<http://csounds.com>

► SlackBot 4.01

Скрипт *bash* для установки программ из исходника в Slackware Linux.

<http://slackbot.sourceforge.net>

► Wine 1.7.11 (Dev-релиз)

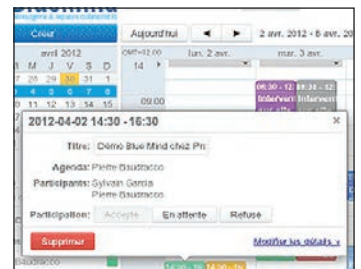
Пакет и библиотеки эмуляции Windows.

<http://winehq.org>

► BlueMind 3.0beta1

Масштабируемая платформа с распределенным доступом для обмена сообщениями и сотрудничества.

www.blue-mind.net



► Теперь с чатом и полным текстовым поиском в *Webmail*.

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Улучшим

Поскольку в этом месяце мы публикуем учебник по Gentoo, я мог бы забыть «Как элитный пользователь Gentoo...», но не стану этого делать, да никогда и не делал: все это интриги бывшего руководства. Суть Gentoo не в элитарности и не в оптимизации опций компилятора до последнего такта CPU — и к дьяволу стабильность. А тогда зачем же он, и чего ради мучиться с установкой вручную? Избитый ответ — «потому что вы это можете»; но Gentoo также демонстрирует одну из сильнейших сторон программ с открытым кодом. Все их части на виду, и если вам нужен дистрибутив ручной сборки с возможностью выбора, то он перед вами.

В дополнение к десяткам (да!) дистрибутивов мы можем создать свой, захотев этого потому, что у нас нетипичное оборудование; или потому, что помешаны на контроле и желаем, чтобы все было по нашей воле; или нам просто интересно узнать побольше о процессе вместо того, чтобы смотреть на черное окно с надписью «нажмите сюда». Но индивидуальная для каждого из нас причина подобного выбора гораздо менее важна того факта, что этот выбор у нас есть, причем для всех, кому он интересен. Еще одна победа открытого кода.

Neil

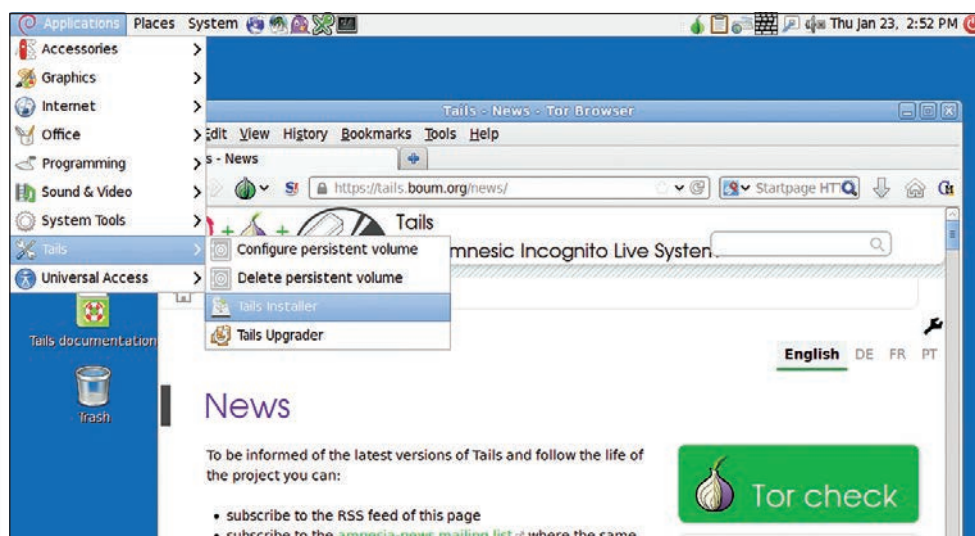
» Важно
ВНИМАНИЕ!

Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

Дистрибутивы Linux

Tails Live



Главная тема этого месяца — конфиденциальность. Помимо отдельных программ, призванных свято охранять эту самую конфиденциальность, у нас есть еще и дистрибутив, посвященный той же цели. Tails Linux основан на Debian, и мы можем быть уверены, что у него стабильный, безопасный

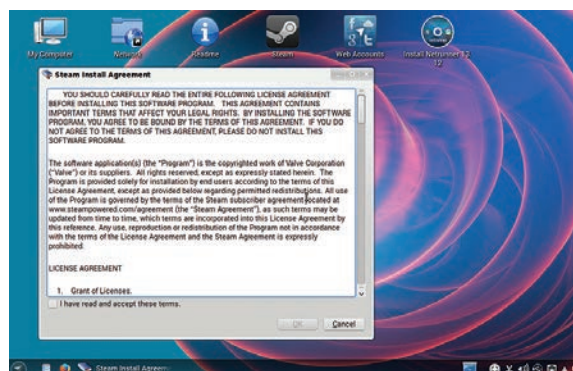
код, но он специально разработан для использования в Интернет. В нем установлен и включен по умолчанию Tor, и Tails настроен на обработку зашифрованной электронной почты. Это портативный live-дистрибутив, так что запускать его с диска или установить на устройство USB.

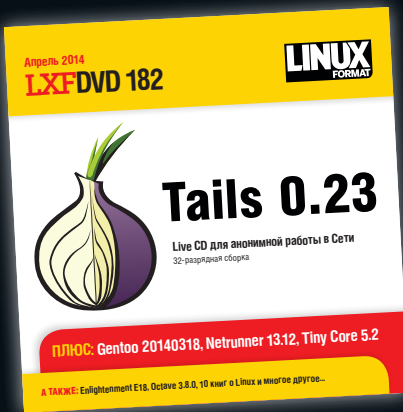
Дистрибутивы Linux

Netrunner (64 бит)

Производных Ubuntu множество, но когда вы упоминаете KDE, большинство вспоминает про Kubuntu. Хотя это прекрасный дистрибутив, Kubuntu слишком преданно придерживается настроек KDE по умолчанию. А между тем появились и другие дистрибутивы, которые радикально изменили настройки KDE, и обычно к лучшему. Поскольку я фанат KDE, мне позволительно будет сказать, что он достоин лучшей настройки, чем по умолчанию. Netrunner — один из таких дистрибутивов, и он показывает, что можно сделать с KDE, если потратить на это время. Однако Netrunner идет еще дальше и способствует развитию KDE, помогая ему улучшаться и добавляя свои нотки. Многим покажется

не менее важным, что Netrunner уже готов запускать Steam: надо всего лишь нажать на значок на рабочем столе для установки.





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.

Из журнала

Конфиденциальность онлайн

ПО, помогающее сохранить вашу драгоценную конфиденциальность, пока вы резвитесь на поляне АНБ.

Учебники

Программы и примеры кода в помощь учебникам этого месяца.

Ответы

Все ваши проблемы с открытым кодом решает этот выпуск.

LXFHotPicks

Brackets

Редактор с открытым кодом, идеальный для веб-дизайнеров и разработчиков интерфейсов.

DevCoin

Вдохновленная этическими принципами криптовалюта для поддержки мириадов проектов с открытым кодом.

GNU Grep

Ищет во входных файлах строки, содержащие совпадения с любым заданным образцом.

Enlightenment

Прекрасный рабочий стол с элегантной внешностью.

GNU Octave

Высокоуровневый интерпретируемый язык, в первую голову предназначенный для числовых вычислений.

PlaneShift

Ролевая игра, перенесенная в виртуальный 3D-мир фэнтези, целиком созданная добровольцами!

Shell_AudioScope

Аудио осциллограф в текстовом режиме, который может записывать звук из `dev/dsp`, `SOX` или демо.

abc2midi

Создает файлы в формате MIDI из текстовых файлов в нотации abc.

GNUCash

Программа персонального бухучета и бухучета для малого бизнеса.

Hnefatafl

Известная также, как Игра Викингов, Стол Короля или просто Tafl.

MKVToolNix

Инструменты для создания, изменения и проверки файлов Matroska в Linux.



И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязанные быть в каждой операционной системе.

Hardinfo Инструмент тестирования системы.

Mentest86+ Проверьте исправность ОЗУ.

Plop Простой менеджер загрузки для запуска операционных систем.

RaWrite Создавайте загрузочные диски в Windows.

SBM Независимый от ОС менеджер загрузки с простым интерфейсом.

WvDial Соединяйтесь с Интернетом через телефонный модем.

Чтение

Книжная полка

The Cathedral and the Bazaar (Собор и Базар) Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

Intro to Linux Руководство для новичков.

Linux Kernel in a Nutshell (Ядро Linux в двух словах) Введение в ядро от магистра хакеров Грега Кроа-Хартмана [Greg Kroah-Hartman].

Справочник администратора Debian Руководство для системных администраторов.

Словарь Linux Linux от А до Z.

Окунитесь в Python Мастер-класс по этому популярному языку.

Руководство по созданию скриптов Bourne Shell Освойте скрипты оболочки.

Расширенное руководство по созданию скриптов Bash Больше информации для опытных пользователей.

Руководство сисадмина Основы работы с системой.



Подкасты



Сезон 6 Серия 1

Настройтесь на новую с иголкой команду экспертов — они заперлись в шкафу и стараются вовсю!

Новости Не надо бояться, но Linux затопил весь ваш дом, холодильники и роутеры рассылают спам, а теперь ваша машина собирается отвезти вас домой из паба... Внимайте, чтобы услышать будущее.

Открытие недели со Steam OS, сделкой Red Hat CentOS и окончанием поддержки Ubuntu 13.04. Плюс мы открыли доступные дистрибутивы Linux, карту преступлений полиции Великобритании, и... что жесткие диски вполне надежны!



Пропустили номер?



Закажите его через сайт www.linuxformat.ru в «ГНУ/Линуксцентр»! Журналы доставляются и в печатной, и в электронной форме, так что с момента открытия браузера до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:



LXF178/179
Январь 2014

250 руб.

- » **Навалился толпой** Революция в экономике?
- » **Начинающим** Взлетная полоса в небо Linux
- » **Qt5** Знаменитая платформа разработки
- » **Штампуют сайты** Да, это реально легко — если с *Drupal*

LXFDVD: Fedora 20 (64-разрядная сборка), Elementary OS Luna, PinguOS, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_178-179/
PDF-версия: shop.linuxformat.ru/elxf_178-179/



LXF180
Февраль 2014

250 руб.

- » **Запах мяты** Mint, симпатия пользователей
- » **Закину на облако** Сервисы хранения данных
- » **OpenStreetMap** Карту мира — своими руками
- » **Платы для Raspberry Pi** Расширяем компьютер-крошку

LXFDVD: Linux Mint 16, Clonezilla 2.2.1, openSUSE 13.1, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_180/
PDF-версия: shop.linuxformat.ru/elxf_180/



LXF181
Март 2014

250 руб.

- » **Все для Raspberry Pi** От плат до программ
- » **Клиенты VoIP** Компьютер, голос!
- » **ARM** Мобильные гаджеты обгоняют ПК
- » **Укомплектуем сами** Машина с вашей архитектурой

LXFDVD: Linux Mint 16, Fedora 20 Live + устанавливаемый, Emacspeak 39.0, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_181/
PDF-версия: shop.linuxformat.ru/elxf_181/

А чтобы не упустить ничего, оформите подписку! Все, кто подписался на печатную версию журнала через www.linuxformat.ru/subscribe/ или www.linuxcenter.ru, получают электронную версию в подарок! На сайте shop.linuxformat.ru вы также сможете приобрести предыдущие выпуски LXF (если успеете).

Подписывайтесь на сайте www.linuxformat.ru/subscribe/

Телефоны отдела подписки:

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



Linux Format ВКонтакте

Вступайте в нашу
группу vk.com/linuxform

На странице LXF ВКонтакте вы найдете:

- » Новости о Linux
- » Статьи из архива LXF
- » Анонс свежего выпуска LXF и часть статей из него
- » Живое общение и онлайн-консультацию по подписке на наше издание

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя_программы-1.0.1.tgz** — тот же файл, что и выше этажом по списку: «tgz» — это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

А ТАКЖЕ: Min-UFO, rftslab и другие обновления

Собран на основе Ubuntu 12.04.3 64bit » Поддерживает UEFI/Secure Boot » Ядро 3.8.0-35

GAINE 5.0

«Black Hole»

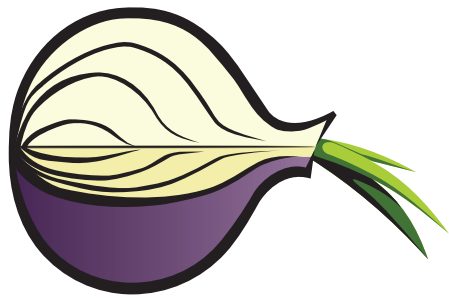
Computer-Aided Investigative Environment

Апрель 2014
LXF DVD 182

LINUX
FORMAT

Апрель 2014
LXF DVD 182

LINUX
FORMAT



Tails 0.23

Live CD для анонимной работы в Сети
32-разрядная сборка

ПЛЮС: Gentoo 20140318, Netrunner 13.12, Tiny Core 5.2

А ТАКЖЕ: Enlightenment E18, Oslave 3.8.0, 10 книг о Linux и многое другое...

Сторона 1

ДИСТРИБУТИВЫ

Centos 20140318 Минимальная установочная система, 32- и 64-разрядные сборки (ISO-образы)
Netrunner 1312 Дистрибутив на базе Debian с рабочим столом KDE, 32-разрядная сборка (ISO-образ)
Tails 0.23 Live CD для анонимной работы в Сети, 32-разрядная сборка (загрузка с LXF DVD)
Tiny Core 5.2 Минимизированный дистрибутив для работы на слабом железе, 32-разрядные сборки вариантов Tiny Core и Core Plus (ISO-образы)

Сторона 2

ДИСТРИБУТИВЫ

CAINE 5.0 Live-система для восстановления данных с различных файловых систем, 64-разрядная сборка (загрузка с LXF DVD)

ДОКУМЕНТАЦИЯ — 10 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

Bash Scripting Подробное руководство по программированию на Bash
Bourne Shell Scripting Начальное руководство по программированию на Bash
Cathedral Bazaar Классический текст Эрика Раймонда [Eric S Raymond] «Сбор и базар»
The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
Dive Into Python Учебник по программированию на Python

Intro to Linux Начальное руководство по Linux
Linux Dictionary Словарь Linux, объясняющий специфическую терминологию
Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом Кроа-Хартманом [Greg Kroah-Hartman]

System Administrators' Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

НОТРИСКИ

abc2midi 20140112 Программа для создания MIDI-файлов из текстовых файлов в нотации ABC
AudioScore 0.21.80 Аудиоцифровщик в текстовом режиме

Brackets Sprint 35 build Редактор HTML
Devcoin 1.0.12 Кошелёк криптовалюты
Enlightenment E18 Среда рабочего стола и менеджер окон
GNUCash 2.6 Бухгалтерская программа для малого предприятия
GNU Grep 2.16 Утилита поиска по тексту
GNU Octave 3.8.0 Вычислительный язык программирования
Infotafel 140117 Древнескандинавская настольная игра
МКУToolBox 6.7.0 Инструментарий для создания, изменения и проверки файлов Matroska
PlaneShift 0.6.0 Игра, многопользовательская ролевка онлайн

**Пожалуйста, перестаньте использовать
имя Аджного Диска озвучивать
с инструкцией, опубликованной
в журнале на стр. 109!**

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аналитический сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных нашей программой или данных. Прежде чем устанавливать какие-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с сайта www.linuxformat.ru с его страницами загрузки.

Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке — например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (скорее всего, в вашей системе присутствует только одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке — метка; затем после адреса SCSI-устройства вы должны указать скорость и 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 придет на помощь утилита *gcombust*. Запустите ее от имени root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажмите на 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 прожечь диск у них. Опять-таки, вам подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.



UNIXFORUM.org



Есть вопросы?
Задавайте!

<http://unixforum.org/>

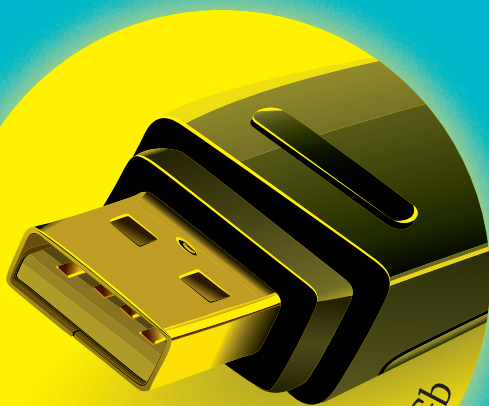
Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

Широкая сеть представительств
в разных городах России
позволит вам оптимизировать
процессы логистики и доставки товара

ПОДРОБНЕЕ О ПАРТНЕРСКОЙ ПРОГРАММЕ:
WWW.LINUXCENTER.RU/PARTNER/



ДИСТРИБУТИВЫ GNU/LINUX НА USB FLASH



Linux Mint 16



Ubuntu 13.10

А ТАКЖЕ
версии для юриди-
ческих лиц —
с лицензионным
договором
присоединения

USB Flash 8 Gb

495 рублей

www.linuxcenter.ru/shop/distros/usb/



UnixEducationCenter

ОБУЧЕНИЕ И СЕРТИФИКАЦИЯ ОС LINUX



ПО ВСЕМ ВОЗМОЖНЫМ УРОВНЯМ:

RHCSA – RED HAT CERTIFIED SYSTEM ADMINISTRATOR

RHCE – RED HAT CERTIFIED ENGINEER

RHCA – RED HAT CERTIFIED ARCHITECT

www.unixedu.ru

+7 (812) 611-15-75 mail@unixedu.ru

Санкт-Петербург, Черноморский пер., д.4

ГНУ/Линуксцентр
приглашает на работу!



ВАКАНСИЯ: Разработчик систем электронного документооборота

ТРЕБОВАНИЯ:

- » Опыт внедрения и сопровождения систем электронного документооборота на базе Alfresco
- » Умение описать бизнес-процесс
- » Знание SQL, понимание принципов построения и функционирования баз данных
- » Знание СЭД, отличных от Alfresco, приветствуется

ОБЯЗАННОСТИ:

- » Разработка систем с нуля, сопровождение
- » Настройка системы в соответствии с требованиями бизнеса
- » Написание инструкций для конечных пользователей

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/



В майском номере Перестрелка дистрибутивов

Этой машине не снести двоих! Мы натравили самые именитые дистрибутивы друг на друга, чтобы выяснить, кто в доме главный.

Освоим openSUSE

Узнайте практически все, что можно, про этот замечательный дистрибутив.

Лучшие инструменты разработки

Тестируем интегрированные среды разработки (IDE), выясняя, как они улучшат ваше программирование.

3D-печать в Linux

Об этой новой технологии сейчас все болтают. Но есть ли чем восторгаться? Разберемся.

Содержание будущих выпусков может меняться — вдруг нас сразит шальная пуля...

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 3000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова, Ирина Шулакова

Редактор диска

Кирилл Степанов

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

ЗАО «Мезон.Ру»

Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. 6/н, лит. Ф
Тел. (812) 462-8383, e-mail: office@ldprint.ru
Заказ 10310

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@futurenet.com

Заместитель редактора Мэтью Хансон [Matthew Hanson]

matthew.hanson@futurenet.com

Выпускающий редактор Крис Торнетт [Chris Thornett]

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоза

[Elfrain Hernandez-Mendoza] elfrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Кент Ельчук [Kent Elchuk], Марко Фиоретти [Marco Fioretti], Дэвид Хейвард [David Hayward], Джо Джордан [Joe Jordan], Джульетта Кемп [Juliet Kemp], Нейл Мор [Neil Mohr], Тэмсин Оксфорд [Tamsin Oxford], Лес Плайндер [Les Pounder], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Ричард Смедли [Richard Smedley], Евгений Балдин, Андрей Ушаков, Алексей Федорчук

Художественный ассистент Карен Льюис [Karen Lewis], Донна-Мари Скразе [Donna-Marie Scrase]

Иллюстрации Шейн Коллиндж [Shane Collinge], Саймон Миддлвик [Simon Middleweek], Элли Уолтон [Elly Walton]

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 01225 442244, email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):

пр. Медиков, 5, корп. 7

Тел. +7 (812) 309-0686

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел./факс +7 (499) 271-4954

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

“GNU/Linux” заменяется на “Linux” в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>



© Linux Format 2005

© Future Publishing Ltd 2005

16+

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

ДИСТРИБУТИВЫ GNU/LINUX НА USB FLASH



Linux Mint 16



Ubuntu 13.10

А ТАКЖЕ

версии для юриди-
ческих лиц —
с лицензионным
договором
присоединения

USB Flash 8 Gb

495

рублей

www.linuxcenter.ru/shop/distros/usb/

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

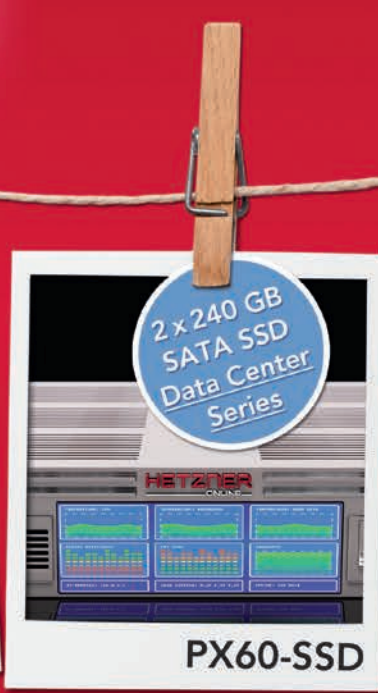
309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center

СВЕЖИЕ РАЗРАБОТКИ!

Новинка!



ВЫДЕЛЕННЫЙ СЕРВЕР PX60

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 2 TB 6 Гбит/с SATA 7200 об/м Software-RAID 1 Enterprise класс
- 20 TB Трафик*

2600

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX60-SSD

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 240 GB 6 Гбит/с SATA SSD Data Center Series
- 20 TB Трафик*

3000

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX70

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 4 TB 6 Гбит/с SATA 7200 об/м Software-RAID 1 Enterprise класс
- 20 TB Трафик*

3000

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX70-SSD

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 480 GB 6 Гбит/с SATA SSD Data Center Series
- 30 TB Трафик*

3700

рублей в месяц + Установка 3700 рублей

ВКЛЮЧЕНО:

- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта



Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выбирайте более чистое будущее вместе с Hetzner Online!

RU.HETZNER.COM



* Нет платы за превышение. При превышении 20 TB/месяц (PX60, PX60-SSD, PX70), 30 TB/месяц (PX70-SSD) скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 80 рублей за каждый дополнительный TB.