

НА ВАШЕМ БЕСПЛАТНОМ DVD
Elementary OS Luna, Hopticks,
PinguOS 12.04 64-бит и сверх того!

LINUX FORMAT

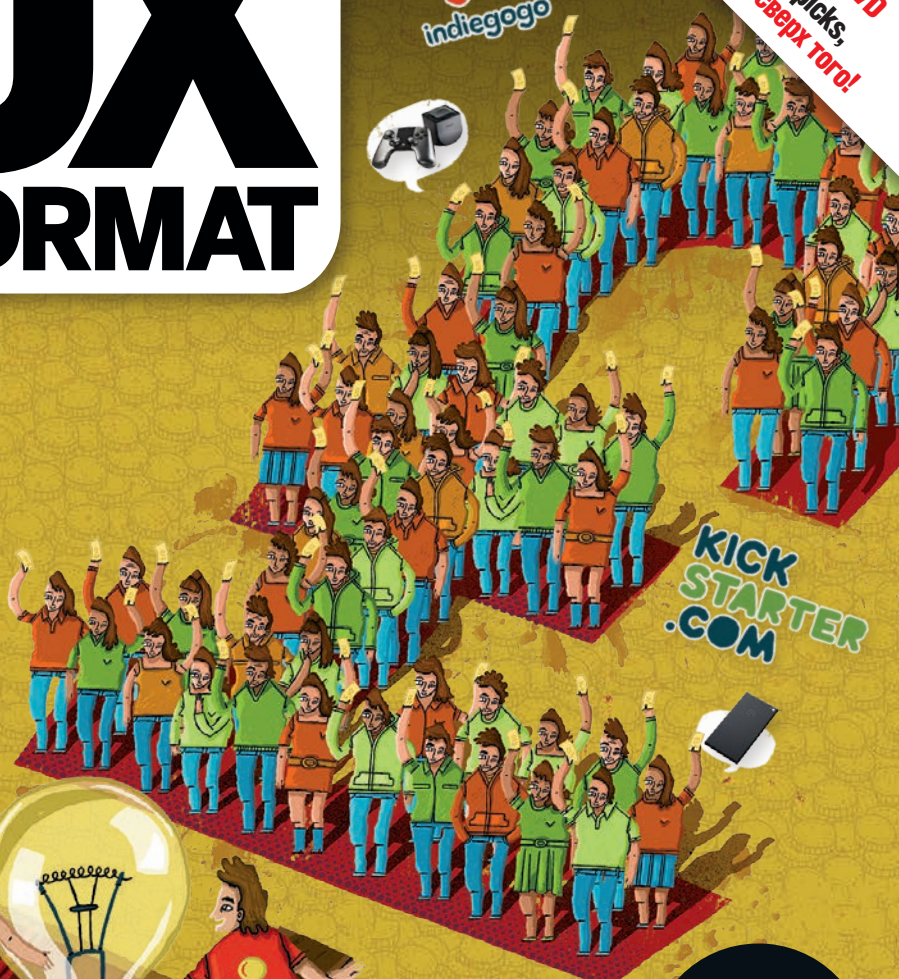
Главное в мире Linux

Январь 2014 № 1 (178/179)

Сбудем мечты всем миром

Как краудфандинг преобразует экономику открытого ПО

- » Получите необходимую поддержку
- » Объедините усилия в Сети
- » Станьте самому себе начальником!



KICK STARTER .COM

ПЛЮС!
Знаменитый Доктор в двойном объеме
с. 56

- Shotwell Организуем отпускные фото с. 66
- Zabbix Отслеживаем все в своей сети с. 84
- Blender Носим маску с изяществом с. 92



Внутри Qt
» Как небольшой графический инструментальный взял штурмом мобильные разработки с. 50



Человек Возрождения
«Многие сегодня пишут больше, чем Диккенс написал за всю жизнь»
Роберт 'r0ml' Лэфковиц о... да обо всем с. 42

Также в номере...

Ubuntu 13.10
Потерял/приобрел/вернул/всегда имел свойства аса? с. 14



Потрошим Nexus
Добьемся доступа root и повеселимся в SuapogenMod с. 22



Qt5
Что новенького у брошенного Nokia сиротки? с. 54



Рабочие столы
Свой личный GUI
» Создадим крутую настольную среду

Построение сайтов
Drupal
» Запросто выкатим мировой сайт Web 3.0

Балансировка нагрузки
HAProxy
» Поборем эффект Slashdot высокой доступностью

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



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

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 отмечаем революцию в финансировании. И мы спросили наших авторов: какой проект они были бы рады увидеть на Indiegogo или Kickstarter?



Крис Торнетт
Мой уже там — это *Hyper Light Drifter* от Heart Machine!



Эндрю Грегори
В своем обычном стиле, я скажу: порт Linux для *Elite: Dangerous*.



Эфраин Эрнандес-Мендоса
Как насчет местного сообщества 3D-печати RipRap в районе Бристолья?



Бен Эверард
После пересечения Африки на деревянном самолете я прошу об одном: внедорожной карте.



Маянк Шарма
Как насчет чего-то вроде Ubuntu Edge, только на сумму \$12 миллионов?



Джонатан Робертс
Моя мечта — создать идеальный пирог с мясом. Какой будет минимальная целевая сумма?



Майк Сондерс
Написав ОС для x86, я теперь хотел бы найти финансирование для ее версии на Forth.



Валентин Синицын
А не найдутся ли желающие продолжить *Dizzy* — с теплой ламповой графикой, но уже для Linux?



Ник Вейч
Я хотел бы откупить права на *Blade Runner 2*, чтобы Ридли Скотт не мог его угробить.



Джульетта Кемп
Я бы запустила проект создания открытого драйвера, в помощь ребятам из Nouveau.



Шашанк Шарма
Исследование космоса следует проводить в складчину!



Нейл Ботвик
Как сказал Грэм, «никогда-то я не успеваю сдать эти ответы вовремя».



Новая экономика

» Тема этого номера, краудфандинг, а точнее — проекты, реализованные таким образом, все чаще появляются в новостях. Об успехах говорят много, но их причина остается за кадром.

Традиционный путь изобретателя известен со времен Кольта и Эдисона. Автор перспективной технической идеи открывает фирму и привлекает деньги т.н. «бизнес-ангелов» — частных лиц или специальных венчурных фондов. Взамен обычно приходится отдавать долю в бизнесе, а единственным критерием успеха становится прибыль.

В случае краудфандинга все происходит «с точностью до наоборот». Привлекая деньги инвесторов, автор обещает вернуть им готовое изделие с заранее объявленными техническими характеристиками. Что и является критерием успеха.

Улавливаете разницу? В первом случае надо вернуть деньги, а во втором — работающее устройство или код. В первом случае все равно, как эти деньги заработаны, а во втором — важен результат.

О том, что экономическая формация, где доминируют крупные корпорации, все чаще рушащиеся под тяжестью собственной бюрократии, вот-вот должна смениться чем-то другим, пишут уже лет 30. Может быть, мы и присутствуем при рождении этой новой экономики?

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



МЕЗОН • ИНТЕГРАЦИЯ

Комплексные решения в области безопасности

Компания «Мезон Интеграция» предоставляет широкий спектр услуг по проектированию, сертификации, монтажу и обслуживанию **охранных систем, систем видеонаблюдения, систем контроля и управления доступом** любой сложности.

Активное сотрудничество с крупнейшими производителями охранных систем позволяет нам устанавливать современное и надежное оборудование для обеспечения вашей безопасности на высоком уровне.

Наши системы безопасности предназначены для:

- /// Паркингов
- /// Зданий
- /// Бизнес-центров
- /// Торговых комплексов и складов
- /// Открытых площадок
- /// Жилых домов
- /// Объектов повышенной секретности
- /// Режимных предприятий
- /// Офисов
- /// Магазинов
- /// Складов

**4-канальная система видеонаблюдения
стоимостью 40 тыс. руб. — в 2 раза дешевле,
чем предыдущие технологии**

ТелеВизард

Многофункциональный сервер общего назначения



ТелеВизард Авто

Самая малобюджетная и высокоэффективная система распознавания автомобильных номеров



Источники видеосигнала

Комплекс eyeSense: видеокамера и объектив. Дополнительное оборудование — ИК-прожектор и климат-защита



si.mezon.ru

Тел. +7 812

309 0686

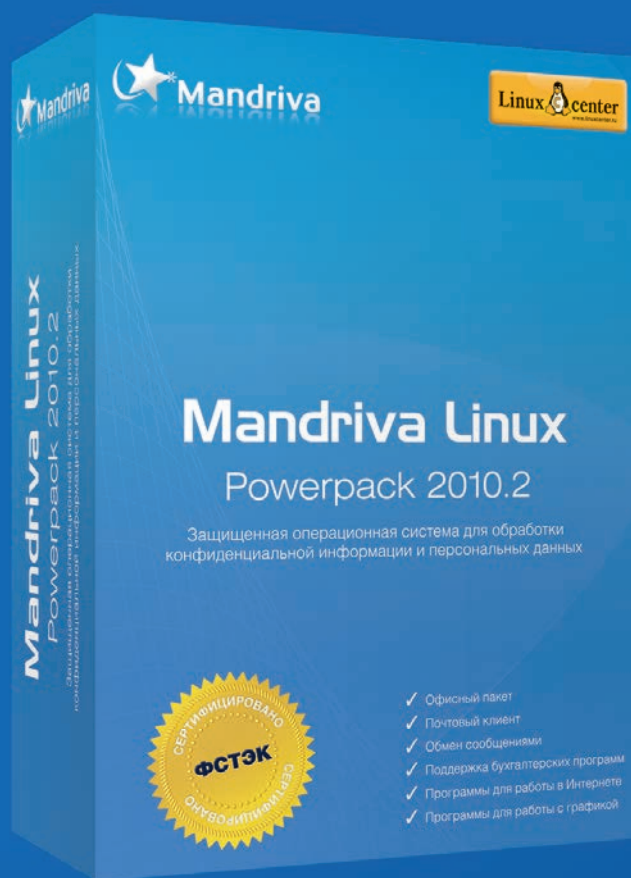
Тел./факс +7 812

640 4990

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

Содержание

Пускай все мы и в канаве, но кое-кто смотрит оттуда на звезды.

Обзоры

Ubuntu 13.10 14

Выхода Saucy Salamander с нетерпением ожидали многие, но не пришлось бы им разочароваться.



› В релиз Ubuntu 13.10 нашли путь новые линзы поиска — но не Mir.

OpenBSD 16

Сумеет ли вечнозеленая открытая версия BSD перейти из серверной комнаты на настольные компьютеры?

Sir, You Are Being Hunted 17

Игра на выживание среди опасных роботов, с добавлением цилиндров и твидя.



› На местность опустилась мгла, пора полночная пришла.

Football Manager 18

Вечно популярный симулятор менеджмента превращает красивую игру в разгул статистики и настроек.

PiGlow 19

Есть такая заповедь: «не трожь индикаторных лампочек». Но кто ж вам теперь помешает завести свои и мигать ими, сколько душе угодно!

Влейте свой голос в

Глас народа

Краудфандинг позволит вам идти своим путем с. 34

Сравнение:
Linux —
новичкам с. 28



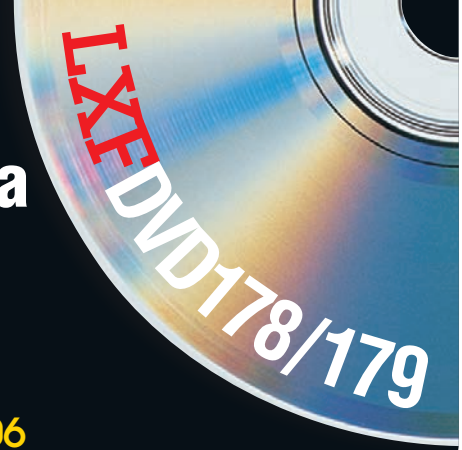
Люди говорят



Мы сейчас на пороге высказывания «О, нужно всех обучить программированию»

Роберт 'roml' Лефковиц 800 лет после Р.Х. и т.п. с. 42

На вашем бесплатном DVD



Elementary OS Luna

» Рекомендуется слушателями подкастов!

PinguyOS 12.04

» Приступайте к Linux

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

Ищите в этом номере

Взломаем Android 22

Покажите своему телефону, кто в доме хозяин.

Мобильная альтернатива 24

На смену Android идет Firefox OS?

Внутри Qt 50

Графический инструментарий, характер нордический.



Пропустили номер?

Узнайте на с. 108, как получить его прямо сейчас!



Учебники

Фотография	
Ах, Shotwell	66
Лидер фотоменеджеров в Linux также и один из простейших в использовании — займемся!	
Сисадмин	
Высокая доступность	68
Управляйтесь с запросами к серверам как профи, по советам профи-сисадмина.	
Хранение данных	
Рулите диском	72
Вычистите хлам, освободив драгоценное дисковое пространство.	
GUI	
Личный рабочий стол	74
Выберите лучшие компоненты и сложите их вместе: Архитектор — вы.	
SSH	
Компьютеры, смирно!	78
Безопасно общаясь через SSH со своими многочисленными ПК, вы их всех построите.	
Продвинутым	
Drupal и его сборки	80
Обращайтесь с Интернетом играючи, подчинив себе эту мощную CMS.	
Сети	
Zabbix бдит	84
Приглядывайте за обстановкой в вашей сети с помощью удобного средства мониторинга.	
Git	
Если вникнуть, то...	88
Мы проломили твердую корку Git и обнаружили, что внутри он мягкий и пушистый.	
3D-видео	
Blender под маской	92
Решаем вопросы сложного композитинга трехмерной сцены.	

Постоянные рубрики

Новости 6

Выпала свежая РОСА, Valve тестирует консоль, телевизор говорит с холодильником, Sailfish ловит Яндекс, вирусы берутся за микрофон, Chrome сбросил тяжести, телефоны становятся мультзагрузочными, а KDE балуется музыкой.

Новости Android 20

SMS-переписка шифруется, у Nokia с Android тайный роман, а Google завел зоопарк роботов.

Сравнение 28

Тестируем пять самых дружелюбных к новичкам дистрибутивов, включая прославленный в Интернете Elementary OS.

Интервью LXF: Об, аж два! 42, 46

Роберт Лефкович полагает, что скоро без умения программировать нельзя

будет считаться членом общества, а **Виталий Липатов** не завязывается на сторонние web-сервисы.

Что за штука... Qt5 54

Легко и быстро создавать приложения и графические интерфейсы для множества платформ — мечта разработчика.

Рубрика сисадмина 56

Доктор отвечает на основной вопрос философии: что за дела со всеми этими заморочками *Wayland/Mir*? А также дарит вам частицу своей мудрости, предлагая *Bleachbit*, инструмент охранения конфиденциальности.

Ответы 96

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик — про проверку параметров SMART жесткого диска, хитроумный код, перенаправление вывода команды, фотографирование

на Raspberry Pi, решение проблем с драйвером, проблемы с разбиением диска, Wi-Fi.

HotPicks 100

Отвечайте горяченько! Лучшие в мире новинки свободного ПО: *Thunderbird*, *XRoar*, *Mars Sim*, *CropGUI*, *DigiKam*, *Fgallery*, *Checkbashisms*, *Dust Racing 2D*, *Chrzzaszcz*, *bd*, *SeaMonkey*.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

Обращайтесь к нам, мы посодействуем. А лучше подпишитесь: тогда потерям и вовсе не быть.

Через месяц 112

За пределами Raspberry Pi; исследуем свежий Mint 16, статические сайты и равноправие в Linux... но не только!



ГЛАВНОЕ Новая Роса » Консоль от Valve » Альянс свободы » Новый телефон
» Звуковой вирус » Chrome полегчало » Гикам на радость » KDE как медиа-центр

НАШИ LINUX

ROSA Desktop Fresh R2

Вышла новая версия российского дистрибутива.

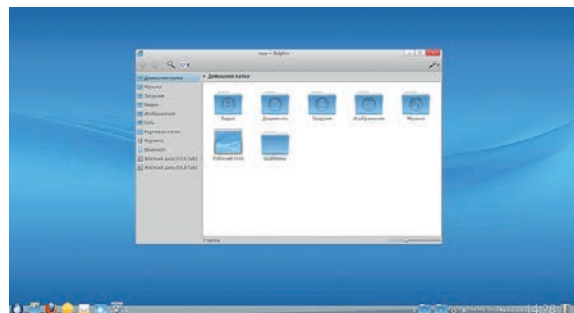
В декабре RosaLab сделала подарок продвинутым пользователям — новый «свежий» дистрибутив.

ROSA Desktop Fresh R2 — плановый айдент-релиз. Линейка «R» предназначена для технически грамотных пользователей, разбирающихся в основных компонентах Linux-систем и желающих получить свежее ПО с широким выбором персональных настроек, без повышенных требований к стабильности системы. «R» развивается как силами специалистов компании «РОСА», так и силами сообщества. Дистрибутив прошел многочисленные этапы тестирования и избавился от «детских болезней», гарантировав поддержку достаточно нового оборудования.

Одно из главных достоинств дистрибутивов семейства Fresh — отличное со-

стояние всех репозиторий и стабильная работа системы в целом. Этого удалось добиться во многом благодаря развитию собственной среды разработки и сборки свободного программного обеспечения ROSA ABF, а также активному применению международных стандартов проверки качества кода. Кроме того, разработчики осуществили переход на новую версию менеджера пакетов, проделали большую работу по адаптации пакетной базы дистрибутива к новым системным компонентам, исправили проблемы с зависимостями и достигли замкнутости репозитория.

Данная версия ROSA Desktop Fresh R2 включает в качестве основного окружения рабочий стол KDE 4. В ближайшее время будут представлены редакции системы с обновленными рабочими столами Gnome



» **Скорость работы ROSA Desktop Fresh R2 значительно повысилась, когда разработчики избавились от использования KDE Plasma.**

и LXDE. Те, кто уже использует ROSA Desktop Fresh 2012 R1, получат обновление до ROSA Desktop Fresh R2 через штатную систему обновлений.

ROSA Desktop R2 будет интересна прежде всего тем, кому нравится менять окружающий мир и меняться самому.



» Рубрику готовил
АРТЕМ ЗОРИН

ДОЛГОЖДАНАЯ КОНСОЛЬ

Паровая машина Valve

Началось тестирование консоли Steam Machine.

В пятницу 13 декабря 2013 года легендарная компания Valve бесплатно отправила первую пробную партию игровой консоли Steam Machine для трехсот энтузиастов, отобранных среди поступивших заявок случайным образом.

К сожалению, вопреки изначальному плану, принять участие в тестировании смогут только пользователи из США. Им отправлено сообщение по e-mail и размещено уведомление в клиенте Steam.

А днем позже Valve анонсировала начало публичного тестирования первого бета-выпуска ОС SteamOS на базе Debian

GNU/Linux 7.1. Для загрузки доступны два установочных образа: **SteamOSInstaller.zip** (960 МБ) с классическим инсталлятором от проекта Debian и **SYSRESTORE.zip** (2,4 ГБ) с уже развернутым архивом системы, пригодным для прямого копирования на USB Flash.

По мнению Valve, дистрибутив Debian является оптимальным вариантом для создания собственных продуктов, адаптированных к решению специализированных задач.

В качестве интерфейса пользователя по умолчанию задействован режим Big Picture, интегрированный в клиент сервиса

» **Внешне Steam Machine напоминает X-BOX, что, вероятно, сделано специально.**

доставки игр Steam (после инициализации запускается клиент Steam). Тем не менее, вопреки ранее поступавшей информации, что SteamOS будет ограничен только режимом Big Picture, в опубликованном варианте ОС оставлена возможность запуска обычного системного окружения на основе Gnome.

Новая игровая консоль — полноценный конкурент Sony PlayStation и Microsoft X-BOX — должна поступить на прилавки уже в начале года.



ИНТЕРНЕТ ВЕЩЕЙ

Создан новый консорциум

Открытая платформа объединит в P2P-сеть потребительские устройства.

Под эгидой организации Linux Foundation создан консорциум AllSeen Alliance, в который вошли такие производители, как LG, Panasonic, Qualcomm, Sharp, Cisco, HTC, D-Link и TP-LINK, заинтересованные в развитии открытой платформы для обеспечения связи между различными типами устройств. В частности, консорциумом будет курироваться разработка универсального открытого фреймворка для поддержки концепции “Internet of Everything” в произвольных продуктах, системах и сервисах. Например, холодильник сможет уведомить смартфон, что надо купить определенные продукты, или вывести на телевизор сообщение о незакрытой двери; электронный замок включит свет при входе в помещение; а стиральная машина уведомит об окончании стирки. С телевизора можно управлять домашней автоматизацией, со смартфона или планшета — быстро организовать вывод на телевизор

или музыкальный центр. В отличие от технологий AirPlay и Miracast, спецификации AllSeen не будут привязаны к вендору и методу соединения.

За основу реализации концепции “Internet of Everything” принят фреймворк AllJoyn, изначально развиваемый компанией Qualcomm, которая передала его

«С телевизора можно управлять домашней автоматизацией.»

под контроль Linux Foundation для разработки независимым сообществом. Все участники консорциума готовы предоставить инженерные ресурсы и технологии для дальнейшего развития фреймворка. Фреймворк распространяется под лицензиями Apache и BSD. Его использование позволит унифицировать средства

взаимодействия потребительских устройств и обеспечить совместимость между оборудованием. Фреймворк предоставит средства для автоматического поиска находящихся в области видимости устройств и сервисов, динамического соединения устройств между собой и организации взаимодействия с расположенными поблизости устройствами, независимо от бренда, транспортного уровня (Wi-Fi, Bluetooth, Ethernet и даже через электро-сеть), платформы или ОС.

Инструментарий AllJoyn позволит создавать приложения и сервисы, способные самоорганизоваться в P2P-сеть, подключаясь к устройствам-соседям через Wi-Fi или Bluetooth. Сеть не привязана к каким-либо централизованным узлам и не зависит от входа в Интернет. Поддерживается автоматическое управление маршрутизацией и передачей сообщений внутри сети, а также возможность аутентификации и шифрования трафика.

СМАРТФОН С ЯНДЕКСОМ

Новый, яркий телефон на Linux

ОС Sailfish от компании Jolla отказалась от Google в пользу Яндекса.

Компания Jolla выпустила в продажу первый смартфон на базе ОС Sailfish. Размер первой партии составил всего 450 устройств.

Смартфон оснащен 4,5-дюймовым экраном (IPS qHD, Gorilla 2 Glass, 720p, 960 × 540), двудерным процессором Qualcomm 1,4 ГГц, поддержкой GSM/3G/4G LTE, 1 ГБ ОЗУ, 16 ГБ Flash, 8-Мпикс камерой с автофокусом и 2-Мпикс фронтальной камерой, Wi-Fi 802.11 b/g/n 2,4 ГГц, Bluetooth 4.0, AGPS, Gionass, microSD, MicroUSB, Micro SIM. Заряд аккумулятора (2100 мА·ч) — примерно на 8 часов беседы. Размер устройства — 131 × 68 × 9,9 мм, вес — 141 г. Стоит оно €399.

По аналогии с CyanogenMod, пользователь сможет заменить штатную Android-прошивку на прошивку с Sailfish. Компания Jolla ведет с производителями устройств переговоры о выпуске моделей с предустановленной ОС Sailfish на базе Android-смартфонов. Подобная трансформация не представляет труда, так

как Sailfish использует стек драйверов от Android, и если устройство может работать под управлением Android, то будет полноценно поддерживаться и в Sailfish.

Платформа Sailfish примечательна использованием графического стека на базе Wayland и Qt5, а также наличием прослойки для бесшовного выполнения приложений, созданных для платформы Android. При этом для установки программ пользователю необходимо вручную загружать пакеты с приложениями или подключать сторонние магазины приложений, такие как Yandex.Store, Amazon Store или F-Droid — поддержка официального каталога Google Play не предоставляется. Интересно, что по умолчанию как поисковая система предлагается Яндекс, а каталог-магазин — Yandex.Store.

Создание приложений для платформы Android при сохранении самобытности Sailfish стало возможным за счет интеграции прослойки Myriad Alien Dalvik, основанной на виртуальной машине Dalvik VM

и оригинале кода Android, что обеспечивает полную совместимость со всеми доступными Android-приложениями без потери производительности.

Использование Яндекса в качестве поисковой системы по умолчанию вселяет надежду, что Россия не останется без этого классного устройства.

» Выбор мобильных платформ растет, и это хорошо, ибо конкуренция двигает прогресс.



ВИРУСНОЕ ИССЛЕДОВАНИЕ

Через звуковую карту — в код

Вредоносное ПО осваивает акустические методы.

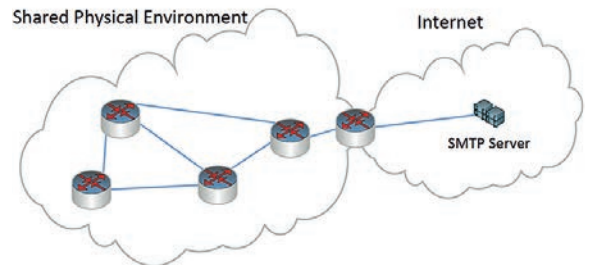
Вирусы для Linux есть, да такие, о которых пользователям и слышать доселе не доводилось. А вот немецкие ученые услышали... сам вирус.

Исследователи из Института информационной и коммуникационной техники имени Фраунгофера (Германия) разработали прототип нового средства коммуникации, пригодного для организации взаимодействия между вредоносным ПО на разных ноутбуках через штатные динамики и микрофоны по звуковому каналу связи. Прототип установил связь между двумя ноутбуками Lenovo T400, помещенными на расстоянии около 20 м друг от друга. В процессе разработки применялись результаты параллельно проводимых исследований по созданию акустических методов передачи данных под водой.

Имитируя работу модема при помощи обычных звуковых устройств, вирусы выявляют наличие поблизости дру-

гих пораженных систем и организуют обмен данными с ними. Вещание ведется в не воспринимаемом людьми диапазоне высоких частот. Скорость передачи в таком канале не более 20 бит в секунду, чего вполне достаточно для передачи таких небольших объемов данных, как перехваченные пароли. Возможный способ блокировать атаку — применение фильтров высокого диапазона частот. Подобные фильтры могут быть реализованы в форме плагина для Linux-подсистемы ALSA (Advanced Linux Sound Architecture).

Продолжение идеи — построение mesh-сети на основе акустического обмена данными, где взаимодействие двух машин можно организовать через промежуточные системы. Практическим применением предложенной системы может быть организация передачи накопленных вредоносным ПО данных от пораженных систем, не имеющих выхода в Интернет,



➤ Система акустической передачи данных идеально подходит для передачи вредоносного кода.

к системам, имеющим канал связи с внешним миром.

Итак, доказана потенциальная возможность создания вредоносных программ, способных передавать вонне данные из систем, не имеющих физического соединения с Сетью. Слухи о выявлении подобных вредоносных систем пока что не подтвердились, но дали повод для размышления о потенциальной возможности использования акустических методов передачи данных.

БУДЬТЕ ПРОЩЕ

Браузер № 2 ускорится

Google запретила размещение в Chrome Web Store усложненных дополнений.

Компания Google сообщила о введении новых ограничений для браузерных дополнений, размещаемых в каталоге Chrome Web Store. Ключевым мотивом изменения правил каталога является желание возврата к первоначальной идее обеспечения максимальной простоты интерфейса браузера Chrome. Сейчас некоторые дополнения перегружены функциями, что приводит к загромождению интерфейса, а иногда и к серьезно замедлению работы. Положение усугубляет то, что некоторые производители ПО без ведома пользователя локально устанавливают подобные раздутые дополнения в процессе инсталляции своих продуктов.

Для сохранения простоты интерфейса решено предоставить пользователю больший контроль над доступными в браузере возможностями и запретить создание раздутых многофункциональных дополнений. По новым правилам, дополнения в Chrome Web Store должны быть нацелены на решение одной конкретной задачи

и быть легко воспринимаемы. Каждое дополнение теперь может быть связано только с одним интерфейсным блоком (UI "surface"), т.е. размещать в интерфейсе только одну кнопку или предоставлять одно действие, привязанное к контексту текущей страницы (иконки в адресной

строке, например, подписка на RSS или показ места на карте). Размещение собственных панелей инструментов теперь запрещается.

При этом дополнения по-прежнему могут предоставлять дополнительные опции, но все они должны быть скрыты за одной видимой кнопкой. Все ранее предоставляемые браузером технические средства разработки дополнений, в том числе возможность создания панелей и все возможности для полного контроля за страницей, будут сохранены: ограничения вводятся только на уровне требований к оформлению дополнений, размещаемым в Chrome Web Store.

Изменение правил уже вступило в силу для помещаемых в каталог новых дополнений. Для существующих дополнений принудительное применение новых правил отложено до июня 2014 года, что дает время разработчикам привести свои продукты в соответствие с новыми требованиями — например, разбить разработку на несколько отдельных дополнений.



➤ Порой Google Chrome становится похож на IE во времена пика популярности последнего.

softline®



Services

Software

Cloud

ИТ-архитектура вашего бизнеса



ОТ ARM К INTEL

Телефон-революционер

На платформе Intel построен смартфон Geeksphone Revolution с поддержкой работы с двумя ОС.

Настольные ПК с двумя и более операционными системами стали нормой, а иногда и необходимостью. Планшеты с подобными возможностями встречаются реже, но тоже уже существуют. Со смартфонами все обстоит совершенно по-другому — на 99,9% моноблоков ставится лишь одна ОС, и не более. Приятным исключением стал смартфон Geeksphone Revolution от испанской компании Geeksphone (одной из первых представившей смартфоны с Firefox OS), который к тому же имеет нестандартную для такого класса устройств начинку.

Geeksphone Revolution собран на аппаратной платформе Intel — Clover Trail+, с процессором Atom Z2560, знакомым по недорогим ноутбукам. Его частота — 1,6 ГГц, а технология Hyper Threading превращает два его ядра в четыре. Что касается ОС, то пока неизвестно, какие именно отобраны производителем.

Он молчит, но можно предположить, что одна из них — Google Android. Предварительный заказ на смартфон Geeksphone Revolution открывается в январе, и тогда же будут раскрыты все интересные подробности о нем. В январе состоится лишь одно крупное мероприятие, посвященное мобильной технике — выставка CES 2014 в Лас-Вегасе, и нетрудно догадаться, где пройдет анонс нового аппарата. Ряд его характеристик известен уже сейчас.

Экран: ЖК-панель — IPS, диагональ — 4,7 дюйма, 960×540 точек, т.е. вполне приличная плотность: 234 dpi. Также Geeksphone Revolution получит слот для SIM-карт, камеру разрешением 8 Мпикс и аккумуляторную батарею емкостью 2000 мА·ч. Об объеме ОЗУ и встроенной флэш-памяти данных пока нет.

➤ **Самое ценное в технологии x86 — это производительность.**



МЕДИА-ЦЕНТР ОТ KDE

Технологии развлечений

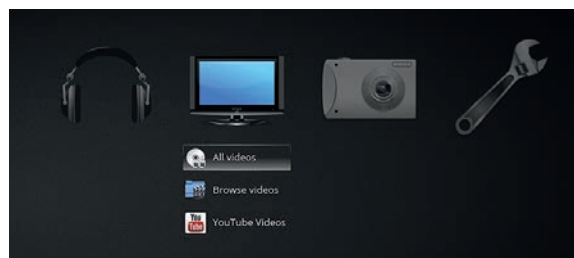
Представлен Plasma Media Center 1.2.

Проjekt KDE анонсировал выпуск Plasma Media Center 1.2 (PMC), унифицированного интерфейса на QML для работы с мультимедиа-контентом на стационарных ПК, нетбуках, планшетах, телевизорах и других устройствах, где запускается KDE.

Предоставляется API для создания плагинов. Поддерживается работа как с содержимыми локальных файловых систем, так и внешних online-сервисов, типа YouTube, Flickr и Picasa. Управление воспроизведением и навигация по контенту осуществляются через клавиатуру/мышь, сенсорный экран или пульты дистанционного управления. Для унифицированного просмотра всего медиа-контента доступны средства KDE Desktop Search.

Ключевые новшества:

➤ Улучшенная работа с музыкальной коллекцией (“All Music”): средства навигации, режимы вывода по музыкантам или альбомам, отображение всей коллекции или элементов по заданной маске.



- Загрузка обложек альбомов и фотографий музыкантов из сервиса last.fm.
- Отображение миниатюр фотографий по содержимому локальных директорий.
- Горячие клавиши управления: для остановки/воспроизведения — пробел, для перехода на следующую или предыдущую композицию — N и Z.
- Управление списками воспроизведения, переключение по серии плей-листов.
- Новые пиктограммы элементов управления, более заметные на темном фоне.

В следующей версии планируются дальнейшие улучшения. **LXF**

➤ **Будущее KDE, возможно, в медиа-контенте.**

Новости короткой строкой

➤ Canonical подписала первое официальное соглашение с производителем о выпуске смартфона на базе Ubuntu Touch.
Источник: www.omgubuntu.co.uk

➤ Проект CyanogenMod получил \$22 млн, в дополнение к полученным в сентябре \$7 млн, от инвестиционных фондов Andreessen Horowitz и Tencent.
Источник: www.cyanogenmod.org

➤ Компания Google разрешила безвозмездно использовать ряд запатентованных ею технологий в проектах, связанных с экосистемой Linux.
Источник: www.openinventionnetwork.com

➤ 17 декабря 2013 года был официально представлен юбилейный 20-й релиз Linux-дистрибутива Fedora.
Источник: www.lists.fedoraproject.org

➤ Патентный суд Германии признал недействительным заявленный на территории Евросоюза патент EP0618540 на методы обработки длинных имен файлов в FAT, применявшийся как рычаг для сбора отчислений с производителей устройств на Linux и Android.
Источник: www.fosspatents.com

➤ Вице-президент Еврокомиссии по вопросам конкуренции, разбирая патентную атаку на Samsung и Google, пригрозил компании Nokia антимонопольным разбирательством.
Источник: www.europa.eu/rapid

➤ Угроза распространения слежки в Сети заставила комитет IETF (Internet Engineering Task Force) предложить разработчикам анонимной сети Tor подготовить интернет-стандарт на базе заложенных в Tor технологий.
Источник: www.technologyreview.com

➤ Приложение CyanogenMod Installer удалено из каталога-магазина Play Store после получения предупреждения от службы поддержки Google.
Источник: www.opennet.ru

➤ Фонд СПО одобрил к использованию ноутбук Gluglug X60 — модифицированную версию ноутбука IBM ThinkPad X60, выпущенного в 2006 году.
Источник: www.fsf.org

➤ Сайт свободной ОС GNU поместил в разделе «Популярные дистрибутивы» предупреждение о неэтичности использования SteamOS.
Источник: www.gnu.org

ПОДПИШИСЬ НА ЖУРНАЛ LINUX FORMAT!

2014

LINUX
FORMAT
Главное в мире Linux



Оформи в редакции
подписку на печатную
версию журнала
и получи в подарок
диск с архивом номеров,
а также подписку
на электронную версию
издания в формате PDF.



Стоимость подписки Годовая — 2280 руб., полугодовая — 1230 руб. без учета стоимости доставки.
Адреса и телефоны редакции Санкт-Петербург, Лиговский пр., 50, корп. 15, тел. (812) 309-06-86.
Москва, Красноказарменная ул., 17, тел. (499) 271-49-54.

Варианты доставки Почтой по России простой бандеролью — журнал доставляют прямо в почтовый ящик
» Почтой по России заказной бандеролью — в почтовый ящик приходит извещение, номера выдают на почте » Курьером
«ГНУ/Линуксцентра» по Москве и Санкт-Петербургу » Курьерской службой СПСР по России » В виде PDF-файлов
для подписчиков электронной версии.

shop.linuxformat.ru

Добро пожаловать в робототехнику!

ScratchDuino

Электронный комплекс на основе свободного аппаратного обеспечения для школ и вузов

ScratchDuino.Лаборатория

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

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

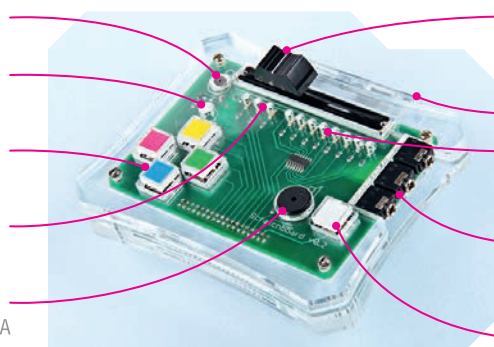
ДАТЧИК ЗВУКА*

ДАТЧИК ОСВЕЩЕННОСТИ*

КРЕСТОВИНА КНОПОК

ЦВЕТНЫЕ СВЕТОДИОДЫ

ДИНАМИК ДЛЯ ВЫВОДА ЗВУКА



ПЕРЕМЕННЫЙ РЕЗИСТОР (ПОЛЗУНОК)*

КАРТРИДЖ ARDUINO

СВЕТОДИОДЫ, УПРАВЛЯЕМЫЕ ЧЕРЕЗ РАЗРЯДНУЮ МАТРИЦУ

РАЗЪЕМЫ ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ ДАТЧИКОВ

КНОПКА*

* РАБОТАЮТ В РЕЖИМЕ ЭМУЛЯЦИИ RISCBOARD, НЕ ТРЕБУЯ ДОПОЛНИТЕЛЬНОГО ПРОГРАММИРОВАНИЯ.

ЦЕНА **5000 руб.**

ГАРАНТИЯ 3 ГОДА, ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ И МЕТОДИЧЕСКУЮ ПОДДЕРЖКУ

ScratchDuino.Робоплатформа

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

ScratchDuino.Робоплатформа

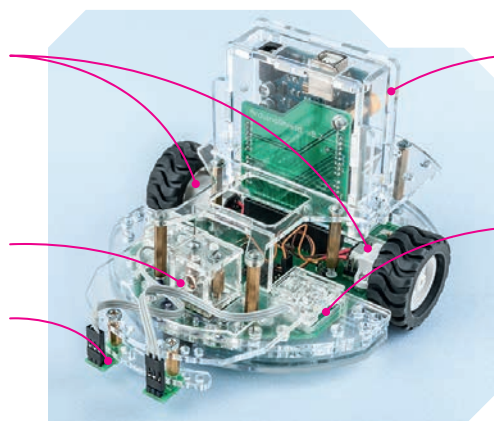
обеспечивает возможность:

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

МОТОР-РЕДУКТОРЫ

ФОТОДАТЧИК

ДАТЧИК ЛИНИИ



КАРТРИДЖ ARDUINO

МАГНИТНЫЙ ЦИФРО-АНАЛОГОВЫЙ РАЗЪЕМ ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ ДАТЧИКОВ БЕЗ МЕХАНИЧЕСКИХ СОЕДИНЕНИЙ И ПАЙКИ*

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

ЦЕНА **10 000 руб.**

ГАРАНТИЯ 3 ГОДА, ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ И МЕТОДИЧЕСКУЮ ПОДДЕРЖКУ

Назначение ScratchDuino

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

Среда разработки

Lazarus (язык Pascal) ■ Scratch ■ Arduino IDE

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

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

WWW.SCRATCHDUINO.RU

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



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Есть ли жизнь без Gnome?

Одной из особенностей Ubuntu, точнее, ее рабочей среды Unity, является редкое «богатство» средств настройки. Здесь и штатный Gnome (!) Control Center (ака Параметры системы), с голым минимумом базовых настроек, и расширяющие его функционал «полуштатный» Ubuntu Tweak Tool и сторонний Ubuntu Tweak. По умолчанию устанавливается только первый, а оба «твикера» по своим функциям перекрываются лишь частично. Если добавить, что часть параметров настраивается только в редакторе *dconf* или консольной утилите *gsettings*, а для юстировки поведения окон может потребоваться менеджер настроек *CompizConfig* — вспоминается пророчество Владимира Попова (автора **LXF**, кстати) о времени, когда количество утилит настройки превысит число параметров.

Разработчики Ubuntu, похоже, увидели приближение рокового рубежа и решили «бороться за упорядочивание стилей», как завещал Великий Мао. А поскольку весь этот зоопарк в значительной мере унаследован от Gnome — начать борьбу с создания форка его центра управления. Чем он будет — пока неизвестно. Можно только надеяться, что он, не став всеобъемлющим монстром типа KDE System Settings, вберет все опции обоих «твикеров», достигнув баланса простоты и функциональности. Ведь пример Cinnamon'a (см. **LXF177**) показывает, что такое возможно. А почему бы просто не интегрировать в Unity Центр управления Cinnamon? Ведь иногда и родителям не запахло поучиться у своих детей. alv@posix.ru

Сегодня мы рассматриваем:

- Ubuntu 13.10** 14
Наш коллективный ум убежден, что Ubuntu повысил осведомленность о Linux больше всех остальных дистрибутивов. 9 лет спустя возникли разногласия. С одной стороны, усилия Canonical унифицировать работу на телефонах, планшетах и ПК. С другой, сторонники Gnome 3.x и Wayland ощущают щелчок по носу.
- OpenBSD** 16
OpenBSD с нами не первый год, в основном как удел серверов. Привлечет ли новый релиз настольных и домашних пользователей? Мы рассмотрели, что там новенького и что оно означает для будущего OpenBSD.
- SYABH** 17
В статье этого номера про краудфандинг мы упомянули игру с ползанием-прятанием-подстрелом, итог успешной кампании на Kickstarter. Ох, как же она нам понравилась!
- Football Manager** 18
Издание-2014 этого невероятно популярного симулятора управления футболом с электронными таблицами идет в Linux. Сохранил ли он свой шарм?
- PiGlow** 19
Еще не поздно стартовать со всякими там цветными гирляндами ЖК-фонариков для вашего дома — есть же



› **Sir, You Are Being Hunted** — настолько длинное название, что многие хорвят сократить его до **SYABH**.

Старый Новый год! Рассмотрим пирмочку к Raspberry Pi, буквально созданную для этого.



› **Mir** не попал в данный релиз, но здесь немало всяких мелких усовершенствований.



› Новый релиз OpenBSD пытался сделать его дружелюбнее, но какой ценой?

Сравнение: Дистрибутивы для новичков

с. 28

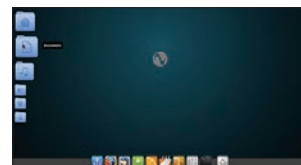
Elementary OS



Pear Linux



PinguyOS



SolusOS



Zorin OS



Внезапно мы заметили, что четыре из пяти наших дистрибутивов имеют 'OS' в названии! И это, несомненно, правильная предпосылка для начинающих линкусоидов: понятно, с чем имеешь дело.

Ubuntu 13.10

Новый релиз — всегда праздник, но увидев в Ubuntu 13.10 все то же, что и раньше, **Шашанк Шарма** малость расстроился.

Вкратце

» Предмет любви и ненависти многих, и все же самый популярный дистрибутив Linux. См. также: Mint, Mageia, openSUSE.

Хотя обычно каждый релиз Ubuntu обсуждается больше, чем многие научные открытия, последнее творение Canonical войдет в историю не за то, что в нем появилось, а за то, чего в нем не стало.

В Ubuntu 13.10 была обещана уйма новых функций, благодаря чему этот релиз стал одним из самых предвкушаемых за долгое время. Однако впоследствии команда Ubuntu навлекла много шума, решив, что столь долгожданная система *Mir* — создаваемая Canonical с нуля, на замену устаревшей *X.org* — в релиз 13.10 не войдет.

В итоге, Ubuntu 13.10 выглядит жертвой жесткого расписания релизов, ничем особо замечательным не отличаясь от своего предшественника.

Применительно к нему очень уместным кажется самый популярный акроним 1990-х: WYSIWIG [англ. сокращение «что видишь, то и получаешь»]. Помимо ядра Linux 3.11, обновленного Unity, необходимых исправлений и улучшения производительности, все прочие новшества — из ряда тех, что можно увидеть и прочувствовать.

Для начала, создать и войти в учетную запись в облаке Ubuntu One теперь можно прямо в ходе установки.

Однако самым обсуждаемым, помимо всего, что связано с *Mir*, стало добавление новых Областей [Scopes] в Dash. Как и прежде, с помощью Dash вы сможете разыскивать приложения, мультимедиа



» Software Center остается одним из главных подарков Ubuntu всему человечеству.

и файлы, но теперь, с появлением Областей, задействуются и данные более 50 online-источников, включая Amazon, eBay, Etsy, Wikipedia, The Weather Channel, SoundCloud, DeviantArt, Foursquare и некоторые проекты Google, такие как Google Drive, Google News и Google Books.

Из любой области?

Имея в своем распоряжении все эти Области, Dash превратился в неиссякаемый источник всезнания. Стоит лишь ввести название своего города, чтобы узнать прогноз погоды, или название альбома или продукта, чтобы найти предложения в Amazon или статью в Wikipedia.

Можно также и ограничить результаты поиска, выбрав только те Области, которые

вы хотите задействовать в данный момент: например, искать только в Google News и Wikipedia.

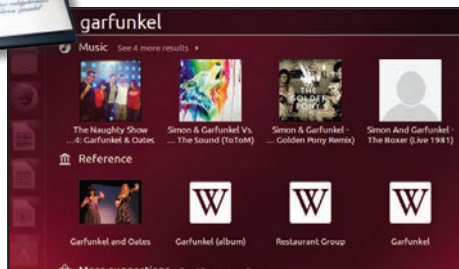
Если выбирать и убирать нужные Области вам лень, можете использовать в запросах префиксы, указывающие на желаемый источник. Так, с префиксом **news:** или **wiki:** будут отображаться только последние новости или статьи в Wikipedia с искомым словом. К сожалению, префиксы доступны не для всех областей, и если одни являются интуитивными и самоочевидными, вроде **books** для Google Books, **wiki** для Wikipedia, **news** для Google News, то же с видео, фото, музыкой, справкой и т.д., то таковые есть не во всех сервисах. Даже не пытайтесь ввести **finance:** или **business:** в надежде поиска на Yahoo Finance.

Dash идеально справляется с поиском по одному слову, но подводит, если сделать запрос длиннее. То есть на слово 'Мадонна' вы получите исчерпывающую информацию, а вот на 'Дэвид Кэмерон', увы, нет, независимо от ваших политических убеждений.

При обработке запроса из нескольких слов Dash ненадежен, хотя иногда и справляется. Как и в Интернете, вы можете заключить искомые слова в кавычки для получения более точных результатов.

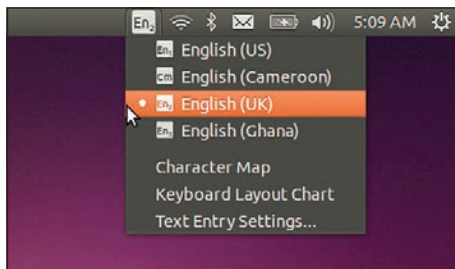
К сожалению, даже задействовав все Области, порой трудно отыскать приложение, которое вы хотите установить. По факту, в ходе наших тестов, выяснилось, что для их поиска приходится специально выбирать Область Другие предложения. В противном случае

Свойства новскидку



Умные области

Dash был существенно переработан изнутри и теперь включает кучу новых областей для поиска с помощью более чем 50 интернет-источников.



Мелкие поправки

Изменения в релизе носят скорее косметический характер: например, на панели состояний добавлена иконка переключения языков.

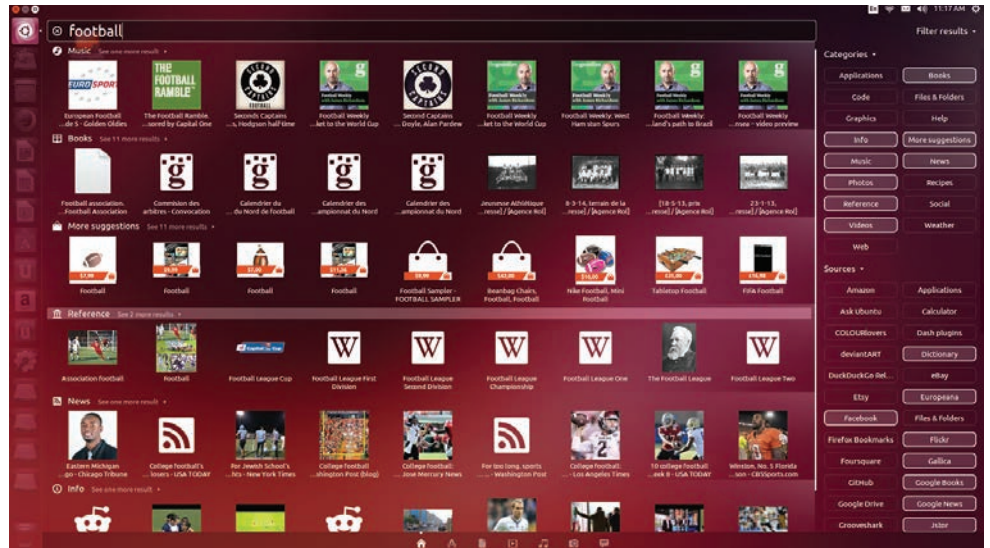
не установленные приложения в результатах не отображались, даже если ввести точное название.

К счастью, Dash по-прежнему отдает приоритет результатам локального поиска, а значит, установленные приложения и недавно используемые файлы и мультимедиа будут отображаться в первую очередь. Результаты по Областям размещаются под соответствующими заголовками, такими как Справка, Музыка или Новости, но на вид все равно очень перегружено.

Щелкнув на любом из результатов поиска, вы получаете более детальную информацию о нем, а также его фрагмент, если речь идет о продукте на Amazon, статье Wikipedia или в новостях. Если это то, что вы искали, можете перейти на страницу оригинала.

Другие промахи

При наличии в релизе всех этих дополнительных интернет-источников для Dash, создатели дистрибутива остались глухи к опасениям по поводу конфиденциальности, высказанным пользователями в связи с желанием Canonical включить результаты поиска на Amazon в релиз 13.04.



➤ Для перехода в нужный раздел результатов нельзя использовать список категорий справа.

только отключить отслеживание активности приложений полностью, а не выборочно, как это было возможно ранее.

Кроме того, в планах по части отказов значился и переход Ubuntu на *Chromium* в качестве браузера по умолчанию; тем не менее, пока что эта должность сохраняется за *Firefox*.

Mobilebuntu

На первый взгляд, Ubuntu 13.10 — первый релиз, содержащий всю функциональность мобильной платформы. Помимо нее самой, это включает и ряд мобильных приложений, таких как веб-браузер, календарь, часы, погоду и калькулятор.

В основе мобильной платформы Canonical лежит их новый и весьма противоречивый графический сервер *Mir* (см. *С небес на Землю*, внизу). Он уже внедрен в мобильной версии Ubuntu 13.10, в настольной же его пока нет, хотя он и был заявлен среди главных целей этого релиза.

В планы Canonical входило включить в 13.10 *XMir*, переходное звено между традиционным графическим сервером *X11* и *Mir*, а также перевести в режим

нейтрализации неисправности некоторые системные компоненты, чтобы отказаться от них в 14.04. Если бы все шло, как задумано, переход на *Mir* в Canonical планировалось завершить в Ubuntu 14.10.

К сожалению, планы Canonical немного пошли вразнос, что интересно, не столько из-за неготовности самого *Mir*, а скорее *XMir*. Разработчики Ubuntu, в качестве одной из самых важных нерешенных проблем, сослались на недоработку системы поддержки различных мониторов.

Из сотен дистрибутивов Linux, критикуемых и прославляемых, ни один не способен так поляризовать сообщество, как Ubuntu. Сколько бы ни кричали на каждом углу о его недостатках, он, как и прежде, остается в числе самых популярных дистрибутивов, когда-либо созданных, и навряд ли будет чем-то вытеснен в ближайшее время (сколько бы его ни обгонял Mint на странице Distrowatch).

Но при всем при том, нет никаких особых причин обновляться до Ubuntu 13.10, тем более, что обновления безопасности для Ubuntu 13.04 будут выпускаться еще несколько месяцев. **LXF**

«Dash превратился в неиссякаемый источник всезнания.»

И хоть в Canonical заявляли, что не хранят никакие личные данные пользователей, при каждом использовании Dash ваши поисковый запрос, страна, местонахождение, источники и Области, активные в данный момент, передаются в компанию.

Более того, в обновленных настройках Безопасности и Личных данных Ubuntu теперь нет вкладки Недавно использованные [Recent Items]. А с ее помощью можно было следить за историей запуска приложений. Пользователям теперь остается

С небес на землю

Графический сервер — важнейшая составляющая рабочего стола. На смену устаревающему *X.org* Canonical создает абсолютно новый *Mir*. Делается это с тем, чтобы реализовать преимущества аппаратного ускорения и гарантировать стабильную работу Ubuntu на разных устройствах. Разработчики также делают акцент на богатых возможностях обработки ввода в *Mir*, с поддержкой жестов на сенсорных экранах.

Mir уже доступен на устройствах Ubuntu Touch, а вот путь к настольной версии оказался длиннее. Чтобы ускорить переход на *Mir* на настольных ПК, разработчики используют *XMir*. В его основе — *X.org*, а *Mir* служит только для цветопередачи.

Именно поэтому, из-за недостатков *X.org*, многим разработчикам трудно оценить преимущества *XMir*. Задержка с появлением *Mir* в 13.10 также стала причиной гнева разработчиков, огорченных отказом Canonical от *Wayland*.

Проект *Wayland*, автором которого является разработчик AIGLX Кристиан Хогсберг [Kristian Høgsberg], должен предоставить протокол графического сервера, подходящий для композитных рабочих столов. Доводы о неуместности дублирования усилий на *Mir* и риске фрагментации, будучи торпедированы заявлениями Canonical о том, что *Wayland* своей цели не достигнет, убедить сообщество не сумели.

LINUX FORMAT Вердикт

Ubuntu 13.10
 Разработчик: Canonical
 Сайт: www.ubuntu.com
 Лицензия: GPL

Функциональность	7/10
Производительность	8/10
Удобство использования	8/10
Документация	9/10

» Так себе релиз; хоть и впервые с поддержкой Ubuntu Touch, но и она не очень впечатляет.

Рейтинг **7/10**

OpenBSD 5.4

Производная от операционной системы BSD давно прижилась на серверах, но сгодится ли она для настольного компьютера? Выясняет Лес Паундер.

Кратце

» Надежная ОС для опытных пользователей Linux и системных администраторов.

Когда вы последний раз пользовались BSD? И вообще, пользовались ли? Довольно давно мы делали обзор этой ОС на базе BSD, и с выходом последней версии OpenBSD 5.4 пришло время посмотреть, что еще нам может предложить BSD.

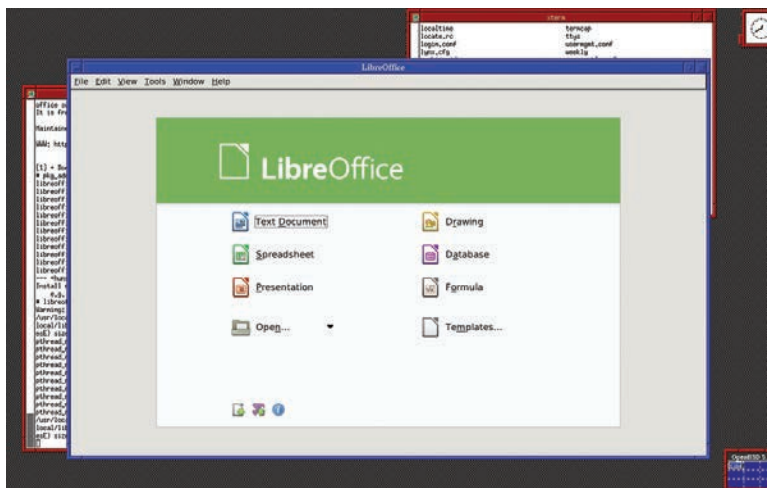
В 1995 году Open BSD ответвилась от NetBSD, и до сих пор ориентирована все на то же: выпуск операционных систем с действительно открытым кодом и безопасностью во главе угла.

С зеркала, предоставленного официальным сайтом, мы загрузили очень легкий ISO-образ — всего на 217 МБ. OpenBSD доступен для многих архитектур процессора; для нашего теста мы загрузили ISO-образ для i386.

Установка — штука хитрая, т.к. графических инсталляторов здесь нет, в отличие от дистрибутивов на базе Fedora и Ubuntu. Текстовый инсталлятор требует кое-какого участия пользователя, но ничего сверх известного многим пользователям Linux, а в большинстве случаев достаточно просто нажать Enter, принимая значения по умолчанию.

После перезапуска компьютера появится хорошо знакомый вам терминал. Войдя, скомандуйте `startx` для быстрого запуска графического стола. При первой загрузке рабочий стол очень скуден: есть только часы и сетка, показывающая доступные рабочие столы. По левому щелчку на рабочем столе появится простенькое меню. Рабочий стол OpenBSD по умолчанию — *F Virtual Window Manager (FVWM)*, и вы можете испытать легкий шок после работы в современных рабочих столах, поскольку в сравнении с другими менеджерами окон *FVWM* выглядит слегка

» Это *FVWM* — менеджер окон по умолчанию. Он, конечно, функциональный, но уж точно не красавец.



» Знакомые приложения рабочего стола, например, *LibreOffice*, можно установить из репозитория OpenBSD.

архаично. Как и в большинстве операционных систем, можно поменять менеджер окон по умолчанию на что-то более привычное. Можно загрузить *Xfce*, *Gnome* или *KDE*. Для своих тестов мы загрузили *Xfce*, и оценили его работу на отлично.

Дистрибутив на каждый день?

Установка любого ПО управляется из терминала с помощью `pkg_add`, а те, кто предпочитает графический менеджер пакетов, могут использовать `pkg_mgr`, который предлагает простой интерфейс *Ncurses*, очень похожий на *Synaptic*.

В нашем обзоре мы исследовали OpenBSD как операционную систему на каждый день и решили установить несколько часто используемых приложений, а именно *GIMP*, *Inkscape*, *Firefox* и *LibreOffice*. Используя `pkg_add`, мы смогли совершенно безболезненно установить все эти приложения и их зависимости.

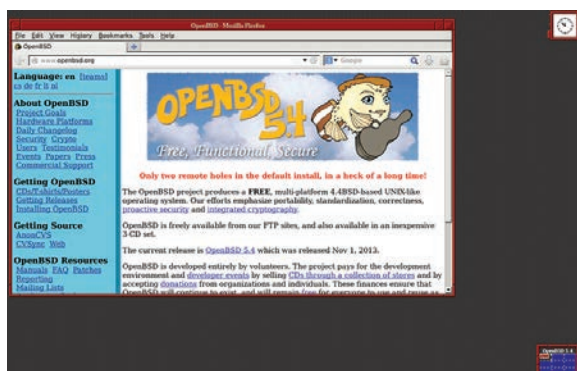
OpenBSD обычно используется для серверов, т.к. обеспечивает безопасность и стабильную базу для создания окна в Сеть. Сходство терминалов BSD и Linux позволяет сделать процесс обучения более гладким, а адаптацию для рьяных системных администраторов — более простой.

OpenBSD вполне пригодна как настольная ОС, и вы найдете здесь много готовых к установке знакомых приложений; рекомендовать ее мы бы не стали, но отчего

и не попробовать, если вдруг возникнет тяга к приключениям.

Возможно, с виду она менее приглядна, чем другие ОС, хотя недостатки внешности восполняет безопасностью и стабильностью. Но новым пользователям OpenBSD 5.4 вряд ли подойдет, и первым же камнем преткновения станет сложный инсталлятор. Если вы только начинаете, то более дружелюбные графические инсталляторы, которые идут с *Mint* и *Ubuntu*, окажутся более подходящим вариантом.

Учитывая вышеизложенное, делаем вывод: если вы опытный пользователь Linux и заинтересованы подтянуть свое знание UNIX, OpenBSD 5.4 станет полезным опытом. **LXF**



LINUX FORMAT Вердикт

OpenBSD 5.4

Разработчик: The OpenBSD Project
Сайт: www.openbsd.org
Лицензия: BSD, ISC, ATU

Функциональность	7/10
Производительность	8/10
Простота использования	6/10
Документация	8/10

» На столь ценный обучающий опыт не жаль потратить время, но это не ОС на каждый день.

Рейтинг **7/10**

Sir, You Are Being Hunted (альфа-версия)

Грэм Моррисон поглощен созданной на пожертвования пользователей игрой хитросплетений, грозных роботов и постоянных обновлений.

Вкратце

» В этой игре на выживание изобилие хитрости и стратегии. Тактика сходна с *Thief*.

Э то не ужасик на выживание. Иначе злые роботы были бы похожи на зомби, дело бы шло кромешной ночью, и вашей задачей было бы придумать побег из темницы, скрытой посреди леса под старым домом в голландском колониальном стиле. Вместо этого вы оказываетесь в случайно сгенерированном пасторальном ландшафте, объединяющем Северный, Юго-Восточный, Западный и Центральный острова; этот архипелаг завален каменными шарами и обломками некоего устройства, раскиданными повсюду в итоге неудачного эксперимента.

По счастью, каким бы ни был эксперимент, обломки накалены так, что дымятся, и ваша задача — по поднимающемуся дыму находить их на местности.

Тут появляются роботы. У них дурацкие роботоподобные голоса и часто повторяющиеся шутки. Это важно, потому что если робота слышно, значит, вы подошли слишком близко. Лучшей тактикой будет избегать этих роботов, потому что, заметив вас, они принимают застрелять.

Роботы имеются разных видов, но симпатичных среди них нет. Если вам повезло найти ружье и кое-какую амуницию, можно сразиться с толпами роботов, но это неразумно, ибо шум привлекает других роботов; хотя некоторые из них висят на воздушных шарах, красный лазер из их глазок-бусин виден за полкилометра. Гораздо лучше швырнуть что-нибудь



» Двигайтесь очень, очень тихо, сэр, на вас идет охота... это роботы.

в другую сторону, отправив роботов в погоню за ложной приманкой, а самим в это время перескочить забор, схватить дымящийся обломок и убежать.

Борьба за пропитание

Для восполнения своей энергии нужно искать, готовить и съедать пищу, причем делать это регулярно, иначе вы дойдете до истощения. Будьте осторожны, готовя на открытом огне: это привлечет внимание честной компании. Можно пользоваться и другими объектами, от бутылок до динамитных шашек, а также тряпками и медикаментами для обработки неизбежных травм.

Искусственный интеллект робота делает игру очень увлекательной и столь же сложной. Роботы, например, примечают даже потревоженных птиц, причем могут подражать любым услышанным звукам и часто берутся за дело вместе, вытряхивая вас из чащи деревьев или из скопления построек. Заставать их врасплох при обходе, который вы обязаны делать часто — вот хороший элемент стратегии, если вы пытаетесь разобраться, куда они направляются дальше.

Вам придется-таки поползти по кустам среди осеннего уныния стилизованного английского пейзажа, а роботы придадут этому занятию зловещий оттенок. Прибавьте сюда ограничение возможности сохранения — и почувствуете напряженность игр старой школы, столь редкую в современных играх. И хотя игра еще на альфа-стадии, мы нашли ее весьма увлекательной. **LXF**



Свойства навскидку



ИИ робота

Искусственный интеллект этих цилиндрических сельских охотников делает игру рискованной.



Сверхчуткость

Вспугнутые птицы предупредят преследователей о вашем местоположении, и прятаться необходимо.

LINUX FORMAT Вердикт

Sir, You Are Being Hunted

Разработчик: Big Robot
 Сайт: www.big-robot.com
 Цена: £ 14,99

Игровой процесс	9/10
Графика	7/10
Долговечность	8/10
Оправданность цены	7/10

» Накал страстей и увлекательность становятся лучше с каждым обновлением.

Рейтинг 8/10

Football Manager '14

В погоне за победой в премьер-лиге Гэри Уокер забывает друзей: Лионель Месси управляемых симуляторов прибыл в Linux. Прямо из Сети!

Вкратце

» 10-й выпуск жутко популярной и чертовски захватывающей серии симуляторов управления футболом впервые воспроизводится на Linux.

Все свелось к последнему дню сезона. Победа над «Астон-Вилла [Aston Villa]» принесла бы «Тоттенхэм [Tottenham]» корону чемпионов премьер-лиги в мой первый сезон как менеджера. Еще немного, и непримиримые соперники «Челси [Chelsea]» выиграли бы титул.

Пока запах сгоревших тостов витал по квартире, а грязная посуда стояла позабытой, мы начали игру с центра поля. Но нападающий «Сперс [Spurs]» Роберто Соладо [Roberto Solado] поддался волнению и был удален за подкат двумя ногами.

К счастью, моя команда «Сперс» — парни крепкие. Когда помощник менеджера Штеффен Фройнд [Steffen Freund] пролаял совет, я велел моим взволнованным подопечным продолжать атаку. Эрик Ламела [Erik Lamela] загнал нас за 11-метровую отметку, но в конце матча Вилла сравняли счет, и запахло переходом титула к «Стемфорд-Бридж [Stamford Bridge]».

Переключаюсь на более решительную расстановку, щелкаю по окнам «форсировать» и «подключить фланги», но на последней минуте Насер Шадли [Nacer Chadli] забивает победный гол. Быстро ударяю по кнопкам «терять время» и «меньше рисков», и «Тоттенхэм» — чемпион.

Эти простые тактические строительные блоки отражают крупнейшие изменения в *Football Manager 2014*, 10-й из серии игр, популярной и среди любителей электронных таблиц, и среди футбольных фанатов, и первой для Linux. Тактическая система радикально переделана. Старые шкалы, где ширина и темп добавлялись щелчками, заменены на предустановленные сочетания, и теперь совет вашего помощника может выполняться по одному щелчку, без выхода из режима панорамы.

Это дало два эффекта: игра стала более плавной и быстрой, но упрощенной



» 3D движок игры в *Football Manager 2014* хорош как никогда.

до абсурда, и требует меньше контроля. Советы помощника идут один за другим, но часть их — нелепые быстрые поправки для сложных сценариев. Ускорение игры чревато случайным щелчком на следующем совете. Некоторые наблюдения вашего №2 могут казаться нелогичными или противоречат друг другу. На одном из собраний команды он даже рекомендовал себе на замену более грамотного работника.

Красивая игра

В этой версии больше способов передавать полномочия, и можно закончить сезон за пару ночей, отключив телефон и дверной звонок. Улучшены 3D-движок, электронный интеллект и анимация спортсменов. Иные проходы с мячом выглядят невероятно реалистично, и если вы не ловили себя на пинании воздуха, когда прорыв увенчивается голом, или на ругательных выкриках фигуркам в майках цветов своей команды, то вы супер-флегмат.

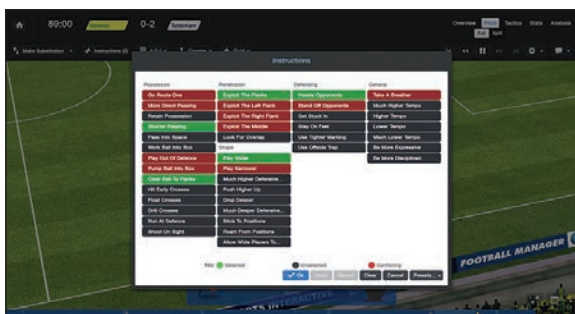
Есть раздражающие глюки на поле: защитники, предупрежденные о слежении за опасным нападающим, могут преспокойно созерцать его прогулку в своей зоне пенальти и забивание гола. В нашей бета-версии были странности: один из защитников напал на моего вратаря Хьюго Лорриса [Hugo Lloris], заблокировал его удар и с наслаждением пробил себе же в ворота.

Взаимодействие более изощрено, чем когда-либо; так, игроки жалуются, что тренировка слишком проста и им нужны новые вершины; можно попросить капитана перекинуться словом с неискушенным

новичком, а менеджеры и бывшие игроки стоят в очереди, чтобы высказать мнение о вашем прогрессе. На пресс-конференциях все еще скучная рутина, хотя можно свалить участие на помощника; новое правило финансового Fair Play от FIFA и рост роли директоров в футболе вполне соответствует реальности.

С сохранением в облаке и FM Workshop игра стала небывало притягательна. Классический режим и серия «Вызовов» с титулами а-ля «Непобедимые» или «Финансовый шторм» предлагают скорейший результат. Приход этого игрового гиганта в Linux — повод для праздника не хуже, чем победа в гостевом матче на стадионе в Эмиратах. Но не забудьте включить домофон, когда ваша команда «Бат-Сити [Bath City]» выигрывает Лигу чемпионов. LXF

» Блоки по щелчку — это новинка тактической системы *FM14* вместо унылых ползунков прошлых версий.



LINUX FORMAT Вердикт

Football Manager

Разработчик: Sports Interactive
Сайт: www.footballmanager.com
Цена: £ 34,99

Игровой процесс	9/10
Графика	7/10
Долговечность	8/10
Оправданность цены	8/10

» Не революция, а лишь эволюция от симулятора управления чемпионатом, но его выход для Linux — настоящий триумф.

Рейтинг **8/10**

PiGlow

Ослеплен ли Лес Паундер блеском новой платы-приставки от Pimoroni?

Вкратце

» Добавочная плата к Raspberry Pi с 18-ю управляемыми светодиодами. См. также: Pi Lite или Pi Matrix.

Pimoroni, компания из Шеффилда, производящая аксессуары для Raspberry Pi, весь год находилась в зените славы, благодаря популярности их корпуса PiBow среди владельцев Raspberry Pi, и в центре внимания прессы, благодаря корпусу PiBow toxic. Но крутые коробочки — это не все, на что они способны, ведь Pimoroni также успешно финансировали кампанию Kickstarter по созданию аркадного мини-автомата, PiCade. А совсем недавно запустили последнюю новинку, PiGlow.

PiGlow — это добавочная плата, как раз под порты GPIO (General Purpose Input Output) вашего Raspberry Pi. В ней 18 ярчайших светодиодов: по три красных, оранжевых, желтых, зеленых, синих и белых. PiGlow необычайно просто устанавливается: нужно просто «насадить» ее на контактные штырьки. Как вы, возможно, и предполагали, PiGlow полностью подходит под корпус Pimoroni PiBow. Хотя PiGlow и занимает GPIO целиком, использует она только 6 портов (а именно, 1, 2, 3, 5, 14 и 17, если вам интересно), а значит, вы можете отвести ее на макетную плату и продолжать использовать остальные.

PiGlow имеет 8-битный 18-канальный ШИМ-чип SN3218 для питания 18 внешних светодиодов и взаимодействует с Raspberry Pi посредством I2C (Inter-Integrated Circuit) через GPIO. Для каждого светодиода можно определить значение ШИМ (широтно-импульсной модуляции) от 0 до 255, а значит, вы можете настроить плавное угасание, приглушив их яркость.

Процесс установки довольно прост и зависит только от выбранного вами способа общения с PiGlow. Среди вариантов: Python, Scrtch, WiringPi и Node.js.

Да будет свет

Для нашего обзора мы использовали великолепные классы Python от Джейсона Барнетта [Jason Barnett], доступные на <https://github.com/Booeeerb/PiGlow>. Чтобы использовать шину I2C на Pi, нам понадобилось установить *python-smbus* и *python-psutil*. После этого мы отредактировали `/etc/modules`, убедившись, что в конце файла стоит `i2c-dev` и `i2c-bcm2708`. Последним пунктом пост-установочной настройки стало редактирование `/etc/modprobe.d/raspi-blacklist.conf`, дабы проверить, что в нем закоментированы `blacklist spi-bcm2708` и `blacklist i2c-bcm2708`. Затем мы перезагрузили Raspberry Pi и создали в домашней директории папку `piGlow`, откуда запустили два скрипта Python: `piGlow.py`, содержащий код, необходимый для работы PiGlow, и `test.py`, который протестирует ее, попросив вас определить яркость каждой группы цветных светодиодов.

Если вы скачаете все классы Python от Джейсона, вам откроется полный диапазон возможностей вашего устройства: например, скрипт CPU способен изменять цвет светодиодов в зависимости от нагрузки вашего Raspberry Pi, и выглядит это совершенно потрясающе.

Раздумываете над тем, зачем вам может пригодиться PiGlow? Ну, для всего, что бы вас ни озаарило, связанного со светом. PiGlow, безусловно, являет собой кардинально новый метод для оригинальной реализации наших проектов.

PiGlow рождает и новые идеи, от выражения настроения при помощи цвета до определения уровня загрузки системы, а благодаря поддержке такого количества языков без труда впишется в любой ваш проект, связанный с Raspberry Pi. Единственным минусом, который мы



» Если PiGlow вам мало, вы, разумеется, всегда можете добавить и светящиеся USB-кабели тоже.

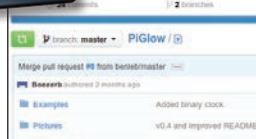
обнаружили, стало то, что он занимает порт GPIO, а значит, будет проблемно использовать его в тех проектах, где GPIO уже задействован, даже если на практике PiGlow нужно всего несколько штырьков и, как уже упоминалось ранее, его можно подключить к макетным платам.

И все же это замечательная штука — еще и при такой низкой цене она безусловно станет выгодным вложением средств для ваших проектов. **LXF**



Свойства навскидку

PiGlow from Python



Языки

PiGlow поддерживает Python, Scratch, WiringPi и даже Node.js, и потому может использоваться в любом проекте.

Порт GPIO

PiGlow подсоединяется ко всем штырькам GPIO, но использует всего несколько, так что можно подключиться к макетной плате.

LINUX FORMAT Вердикт

PiGlow

Разработчик: Pimoroni
Сайт: <http://shop.pimoroni.com/>
Цена: £9

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

» Новая деталь, обеспечивающая вывод совершенно уникальным способом. Срочно добавляйте ее к своему проекту Raspberry Pi.

Рейтинг 8/10



ЗАЩИЩАЮТ ТЕЛЕФОНЫ

Больше безопасности

CyanogenMod обеспечила передачу и хранение SMS в зашифрованном виде.

CyanogenMod 10.2 реализовала прослойку WhisperPush для работы с сообщениями SMS и MMS, с поддержкой протокола TextSecure. Этот протокол и Android-приложение с его реализацией разработаны компанией Whisper Systems и открыты после поглощения фирмы компанией Twitter. TextSecure позволяет шифровать SMS-сообщения.

Штатные возможности обслуживания SMS-сообщений сохраняются, но если система получателя поддерживает TextSecure, отправка производится не через сотового оператора, а по зашифрованному каналу связи. Архив сообщений хранится в зашифрованном виде, исключая чтение сообщений при попадании телефона в чужие руки. Все операции выполняются автоматически и полностью прозрачны для пользователя. Не требуется обмен

» Единственный минус для многих в CyanogenMod — отсутствие FM-радио. Зато безопасно.



ключами или выход online — отправка не отличается от обычных SMS.

Для технических специалистов добавлена опция проверки и изменения ключей идентификации (людям, с которыми уже была зашифрованная переписка, при смене ключа поступает уведомление). Методы шифрования — Curve25519, AES-256 и HmacSHA256. CyanogenMod ввел свой собственный TextSecure-Server, синхронизированный с сервером Open WhisperSystems, обеспечив обмен SMS между пользователями CyanogenMod и Android-приложения TextSecure. Ожидается дополнение браузера TextSecure, для обмена SMS также и с пользователями настольных систем.

Сборки CyanogenMod 11.0 на Android 4.4 пока доступны только для устройств от Google — Nexus 4, 5, 7 и 10.

СЕНСАЦИОННЫЙ СЛУХ

Android — в Nokia?

Nokia работает над созданием смартфона на платформе Android.

Время от времени в сети появляется информация о том, что финская компания Nokia может представить бюджетный смартфон, работающий под управлением ОС Android.

В конце декабря 2013 года пользователь Твиттера @evleaks опубликовал пару рендеров Android смартфона от Nokia,

а журналисты портала The Verge поделились подробностями об этом устройстве.

Nokia разрабатывала данный смартфон как альтернативу линейке Asha. Новинка под названием Normandy усилила бы позиции компании в развивающихся странах. В отличие от простых аппаратов Asha, Nokia Normandy сможет использовать заделы приложений и игр для Android. С учетом продажи Microsoft мобильного подразделения и бренда Nokia, Normandy, видимо, станет первым и последним аппаратом Nokia на зеленом роботе.

По статистике, именно Android сейчас доминирует на рынке «умных» сотовых аппаратов с долей около 80% в общем объеме поставок в третьем квартале.

Однако китайский сайт C Technology, одним из первых опубликовавший информацию о подготовке в Nokia под руководством Питера Скиллмана [Peter

Skillman] Android-смартфона серии Asha, утверждает, что Nokia не намерена развивать это направление: научно-исследовательское подразделение Nokia сейчас занято новыми проектами переносных компьютерных устройств, с поддержкой технологии дополненной реальности и, возможно, гибким дисплеем. Также ведется работа над новыми технологиями беспроводной передачи энергии и генерирования электромагнитной энергии. Решение данных проблем позволит Nokia значительно опередить конкурентов.

По условиям продажи Microsoft, Nokia еще два года не имеет права выпускать мобильные устройства под своим брендом. Поэтому, по мнению китайцев, слухи о выпуске в 2014 году Android-смартфона Nokia беспочвенны: новый бренд раскручивается не мгновенно, да и стоит это недешево.



» Самое смешное, что смартфон уже как бы есть, если верить Интернету...

ЕСЛИ БЫ НЕ APPLE!

Поворотная точка истории

Журналисты выяснили, что Apple подложила Google свинью...

Знаете ли вы, что Стив Джобс [Steve Jobs], представив публике iPhone (а случилось это 9 января 2007 года), буквально ударил под дых команду Google, которая уже два года секретно работала над выпуском своего смартфона? По крайней мере так пишет Atlantic — американский аналитический портал.

Итан Берд [Ethan Burd], тогдашний руководитель команды Android, сказал: «Все знали, что в Apple трудятся над собственным смартфоном, но никто не подозревал, что он окажется настолько крутым». Наблюдая за презентацией Apple, Энди Рубин [Andy Rubin], следовавший тогда на выставку CES 2007 на встречу по делам Android, говорят, буквально остановил машину, чтобы целиком посмотреть выступление Джобса. После чего сообщил коллегам, что Google не будет продавать свой Blackberry-подобный гаджет.

Кодовое имя устройства, уже варившегося в лабораториях Google, было Sooner. Но оно так никогда и не поступило в продажу из-за iPhone. Некоторые утверждают, что девайс был более инновационным



➤ Внешне Google Sooner напоминает мегапопулярный в 2007 году Blackberry.

и функциональным, чем только что представленный iPhone, но имел большую традиционную qwerty-клавиатуру и обычный, не сенсорный экран, из-за чего выглядел куда менее привлекательным и желанным, чем продукт Apple.

«Как потребитель, я был в восторге, хотелось немедленно обладать iPhone, но как инженер я понял, что нам придется

начать многое заново, — говорит Крис де Сальво [Chris De Salvo]. — Наш девайс вдруг резко стал выглядеть приветом из прошлого. Теперь хоть сервис-деск на нем запускай, но Sooner не вызывает такого же желания купить его, как iPhone».

В итоге, после двух лет работы над Sooner, намеченным к продаже уже в конце 2007 года, команде Android пришлось менять вектор разработки на совершенно новое устройство — смартфон с большим сенсорным экраном, способный соперничать с iPhone. Новый гаджет получил кодовое имя Dream, а его запуск был намечен на конец 2008 года.

Однако другой инженер Google, Денни Хэкберн [Danny Hackburn], сообщает, что все происходило несколько иначе. Компания работала сразу над двумя устройствами: Sooner и Dream. Система на обоих была одна и та же, но девайсы были разного форм-фактора. В итоге после выпуска iPhone более классический Sooner, который был похож на те гаджеты, что некогда работали под Windows Mobile, остался за бортом как бесперспективный.

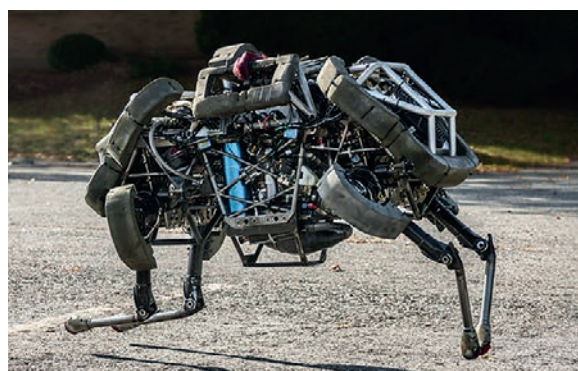
ЗВЕРОПОДОБНЫЕ РОБОТЫ

«Фауна» от Google

Наверное, такие механические собаки и мерещились Рэю Бредбери.

Корпорация Google приобрела компанию Boston Dynamics, разрабатывающую роботов, в т.ч. для Пентагона. Компания из Уолтема, штат Массачусетс, создает машины с потрясающим чувством равновесия, способные обгонять самого быстрого человека,

➤ Четвероногий робот от Boston Dynamics по имени WildCat бежит очень быстро.



преодолевать грязь и снег и попутно переносить грузы. Это восьмая робототехническая компания, приобретенная Google за последние полгода. Теперь к ее механическому зверинцу прибавились BigDog [Большой пес], Cheetaah [Гепард], WildCat [Дикий кот] и человекообразный Atlas [Атлас], пишет *The New York Times*. Google пока не говорила, что будет делать со своей коллекцией, но, вероятно, координировать проект будет Энди Рубин, бывший глава разработчиков ОС Android, назначенный начальником отдела по созданию роботов.

Многие роботы Boston Dynamics были разработаны при поддержке Министерства обороны США, напоминает *The Financial Times*. Google может найти им применение в гражданском секторе — от доставки и работы на складе до ухода за престарелыми. «Будущее выглядит впечатляющим!» — написал в Twitter Рубин.

Вкладывает значительные средства в робототехнологию не только Google. Так, недавно крупнейший интернет-ритейлер Amazon потратил \$775 млн на компанию Kiva Systems ради автоматизации своих центров реализации, а вскоре планирует развивать сеть доставки товаров с помощью беспилотных летательных аппаратов.

Boston Dynamics была основана в 1992 году Марком Райбертом [Marc Raibert], бывшим профессором МТИ. Компания не производила роботов в промышленных масштабах, но работала над расширением возможностей техники и созданием новых технологий. Ее модели роботов всегда привлекали к себе повышенное внимание. Последнее официальное видео компании в YouTube, репортаж с испытаний робота WildCat с двигателем внутреннего сгорания, умеющего бегать и разворачиваться, было просмотрено более 15,7 млн раз. LXF



» Доступ root Разблокируйте свой телефон и/или планшет

Взлом Nexus 5

Верните контроль над своим устройством Android с помощью **Нейла Мора** и TWRP (т.е. Team Win Recovery Project, если вам интересно).



Наш эксперт

Нейл Мор занимается кодированием и играет с компьютерами, начиная со своего первого ZX81 в 1981, и изо всех сил кричит о — или на — Linux все последнее десятилетие.

Разблокировка и вход от имени root на устройство Android является главным условием обретения над ним контроля. В отличие от Apple iOS или Windows Phone 8, где доступ root, и, следовательно, полный контроль, считается погрязанием, над которым ехидно посмеиваются многие владельцы мобильных, в мире Android большинство производителей в конце концов пришли к принятию этого процесса.

Проблема при написании любого руководства в том, что каждое устройство имеет собственный процесс разблокировки и входа под root. В более ранних устройствах это было весьма проблематично. Но, к счастью, после выхода Android 4.0 ICS большинство производителей предлагают один и тот же путь. Мы рассмотрим самое популярное из устройств Android — Google Nexus 5. Хорошая новость в том, что описанная здесь технология будет работать со всеми устройствами Google Nexus, кроме оригинала — Samsung Galaxy Nexus; хотя он в общем-то похож.

У других производителей вы найдете ряд устройств, которые используют тот же самый базовый инструмент разблокировки и заливки прошивок [flashing]. Многие более старые устройства Samsung используют инструмент *Odin*; более новый ряд Samsung Galaxy использует инструмент под названием *Heimdall*. Все они поставляются через сайт CyanogenMod. Подозреваем, что в будущем появятся еще больше инструментов от Samsung, названных именами богов из северного пантеона.

Устройства HTC практикуют немного другой подход, который требует установки нужного драйвера от разработчика HTC, и предлагают официальный путь через сайт www.htcdev.com/bootloader. Для самых последних моделей HTC One этот сайт предлагает файл разблокировки, который вы отправляете на устройство с помощью команды Google Fastboot. Для более старых



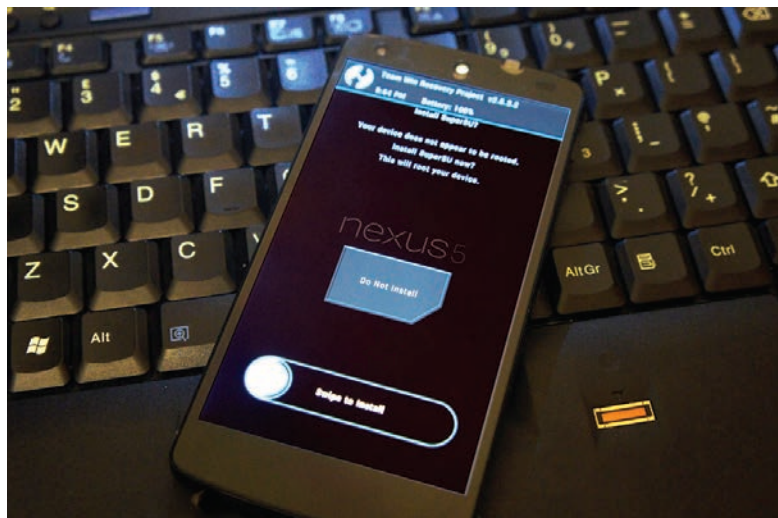
» Когда вы начнете устанавливать и латать индивидуальные программы Android, этот экран будет частым зрелищем.

моделей предлагается поддержка через обновленную систему загрузчика HBOOT, но CyanogenMod по-прежнему предлагает использование оригинального инструмента загрузки Revolutionary. Так или иначе, результат будет один. И последний пример: Sony в своей последней линейке устройств тоже предлагает официальную поддержку на <http://unlockbootloader.sonymobile.com>. Вам пришлют код разблокировки, который вы передаете на устройство через команду Google Fastboot.

Со своей сердцевиной Linux, Android открыт для всех, кто перестраивает свою версию ОС. CyanogenMod остается самым популярным, и сейчас даже предлагает основанную на Google Play программу установки, что является мощным прорывом. CyanogenMod предлагает простую версию Android, но со множеством прилагаемых индивидуально настраиваемых функций. Например, он по умолчанию поддерживает скины. Независимо от того, какую сеть вы используете, он предлагает режим модема через соединение USB, Wi-Fi и Bluetooth. В нем есть встроенная поддержка OpenVPN плюс высококачественного аудиокодека FLAC. И поверх всего этого имеется широкий ряд возможностей для расширенной индивидуальной настройки устройства.

Все это звучит замечательно, однако есть ли здесь недостатки и риски? Как и в любом процессе, который обновляет или заменяет встроенные программы устройства, некий риск существует, даже при использовании официальных инструментов и файлов. В зависимости от вашего устройства, есть также вероятность, что процесс разблокировки уничтожит данные. К счастью, шансы превратить ваше устройство в кирпич (дохлый) невероятно малы, особенно если вы последуете нашим советам. А что до недостатков попроче — они, конечно, имеются.

» Получить доступ root намного проще через TWRP, благодаря его встроенному инструменту rooting.



Бутсы, как у Бэкхема

В силу разных сложных причин, которых мы коснемся позднее, сборки для некоторых устройств могут не поддерживать всего оборудования устройства. Ситуация, когда GPS, Bluetooth и беспроводное или мобильное соединение не работает, довольно обычная. Возможно также, что некоторые кнопки не распознаются, отсутствует аудио, или у батарей в устройстве моментально заканчивается заряд. Хорошая новость в том, что обо всем этом вас предупреждают заранее, и вам не придется брести вслепую. Чем более зрелое у вас устройство, тем больше вероятность, что подобные ошибки уже исправлены.

Сначала — важнейшее ПРЕДУПРЕЖДЕНИЕ: процесс разблокировки уничтожит все данные на вашем устройстве. Поэтому или сделайте копии, или научитесь относиться к этому философски еще до процедуры. Для доступа к устройству Android используется несколько объектов — для начала, драйвер, который дружит с Linux. Также вам понадобятся два инструмента разработки Google, *adb* и *fastboot*. Установите их оба через **apt-get** таким образом:

```
sudo apt-get install android-tools-adb
sudo apt-get install android-tools-fastboot
```

Вам также нужно подготовить свое устройство, не только активировав Developer Mode, но также и переведя его в режим USB debugging [Отладка USB]. Если у вас новое устройство, то меню Settings > Developer Options [Настройки > Опции разработчика] будет скрыто; чтобы его показать, выберите Settings > About [... > Про], переместитесь вниз к пункту Build [Сборка] и стукните по нему семь раз. И он объявит, что теперь вы — разработчик; музыка, туш! В новом меню тоже активируйте режим USB Debugging.

Теперь дело становится нешуточным, поскольку надо перезагрузить Nexus в Fastboot, специальном режиме восстановления его загрузчика. Для этого выключите телефон, нажмите и удерживайте обе кнопки громкости, и он включится. Он должен немедленно запуститься в режиме Fastboot, показав зеленого человечка Android с разверстой грудью... бедолага.

Fastboot быстро, как... Рональдо?

Присоедините Nexus к вашему Linux ПК, и в окне терминала введите следующую строку, Nexus выведет предупреждение, которое стоит прочитать. С помощью кнопки громкости выбирайте Yes, а с помощью кнопки питания — No:

```
sudo fastboot oem unlock
sudo fastboot reboot
```

Nexus перезапустится и удалит систему, затем загрузит новую установку Android. Спасибо родному *Linux Format* за науку, как грохнуть все данные на нашем телефоне! Ну да, это правда, но теперь-то мы расскажем, как войти от имени root и удобно установить прошивку [ROM] для восстановления, которая чертовски упрощает резервное копирование всей системы и заливку индивидуальных прошивок.

По миру Android рыщут, как два огромных зеленых, э-э, андроида, две восстановительных прошивки. Это ClockworkMod и TWRP, сокращение от Team Win Recovery Project. Ребята они порядочные, поэтому для Nexus 5 используйте самую последнюю версию с www.teamw.in/project/twrp2/205. Полный список поддерживаемых устройств находится на www.teamw.in/twrp/view_all_devices. Скачайте файл IMG на ваш Linux-ПК и присоедините к нему Nexus через кабель USB. Полный список устройств для ClockworkMod можно найти на www.clockworkmod.com/rommanager.

Нам надо перепрограммировать устройство еще раз, в режиме Fastboot, поэтому выключите его и включите снова, нажав на обе кнопки громкости. Введите следующую строку в терминал, указав правильный путь и номер последней версии:

Портируйте это

Вы можете поинтересоваться, почему CyanogenMod не может быть доступен для всего сразу? Ведь все это — устройства Android, так почему бы просто не установить версию на старое устройство?

Есть пара наследственных причин. Мы избегаем называть их проблемами — ну, просто именно так обстоят дела. Первая в том, что Android не похож на большинство дистрибутивов Linux, поставляемых со множеством родных драйверов, благодаря чему они работают на ряде устройств. Android OS собирается для конкретных устройств при наличии

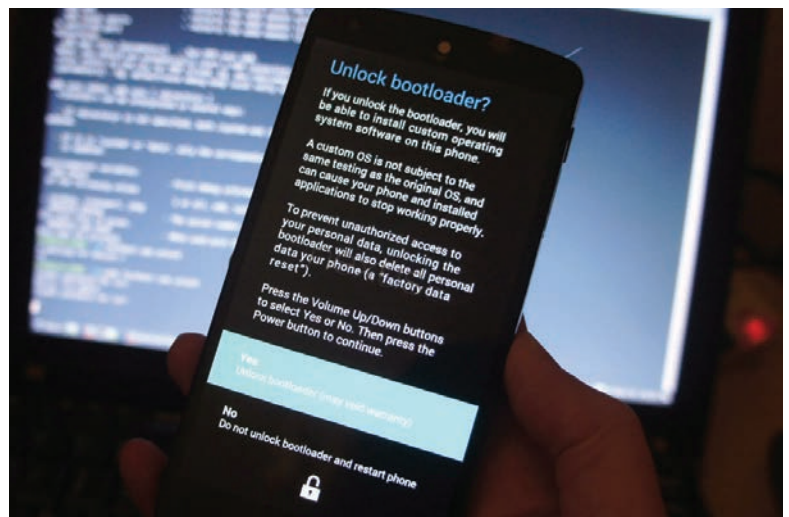
специальных драйверов. Пока разработчик не смиляется и не создаст версию, или пока вы сами не научитесь программировать, это является загвоздкой.

Вторая, связанная с первой, причина — хотя код Android открыт, а производители должны открывать код для драйверов и целых версий Android, это зачастую происходит с опозданием или вообще не происходит. Если у вас есть такое подпорченное устройство, это означает, что отдельные его элементы могут оказаться неработающими: аудио, GPS, беспроводное соединение и т.д.

```
Sudo fastboot flash recovery /home/<username>/Downloads/
openrecovery-twrp-2.x.x.x-hammerhead.img
```

Используйте кнопки громкости для отображения Restart bootloader, а для выбора — кнопку питания. После перезагрузки Fastboot, выберите режим Recovery и загрузитесь в TWRP. Добро пожаловать в ваш индивидуальный восстановительный загрузчик; он очень похож на ClockworkMod, но с более дружелюбным интерфейсом. Очень кратко коснемся опций: итак, Install позволяет программировать образы прошивок и устанавливать обновления в виде ZIP в свою ОС. Backup и Restore позволяют делать резервное копирование и восстановление целых разделов вашего устройства Android. Опции Advanced в Wipe могут оказаться полезными инструментами для решения проблем, если таковые возникнут с кэшем JVM Dalvik или с системным кэшем; в ином случае они могут полностью стереть фабричные настройки.

Одна из причин выбрать TWRP заключается в том, что в нем есть встроенный инструмент Android root. Если вы выберете Root > System, он выяснит, администрируется ли ваше устройство Android. Если нет, он предложит сделать это; сдвиньте ползунок, и Android перезагрузится в администрируемый режим. Однако на этом дело не заканчивается: перейдите в раздел приложений и найдите приложение SuperSU. Оно сообщит, что ему требуется установка его интерфейса пользователя — это можно сделать через Google Play или прямую загрузку. Оба способа хороши, но для прямой загрузки нужна заплатка в архиве через удобный и уже установленный TWRP, и все это делается автоматически. **LXF**



➤ Разблокировка загрузчика — первый шаг к свободе вашего устройства Android, невзирая на пугающее предупреждение.



» Альтернатива! Запрограммируйте свой свободный мобильный гаджет на Firefox OS

Firefox OS: Пишем приложение

Присоединяйтесь к Лесу Паундеру — он покажет, как написать, протестировать и загрузить собственное открытое приложение.



Наш эксперт

Лес Паундер обожает возиться с новыми языками и гаджетами, и ему очень нравится jQuery.

Что такое Firefox OS? В последние несколько лет к основным мобильным ОС относились iOS от Apple, Android от Google и BlackBerry. Ни одна из них не является открытой — даже Android не открыт полностью.

Подлинно открытую мобильную ОС решила создать Mozilla, и это Firefox OS. Она основана на урезанной версии Android, из которой удален проприетарный код и оставлен только открытый.

Mozilla активно продвигает открытое ПО, и Firefox OS — самая свежая мобильная ОС, которая стремится занять эту долю рынка. Попытки создать открытую мобильную ОС были и прежде — например, OpenMoko (http://wiki.openmoko.org/wiki/Main_Page).

Firefox OS отличается от других мобильных ОС максимальной открытостью. Здесь используется обычное ядро Linux, а для построения интерфейса и реализации всего его функциональности используются только открытые web-технологии. На самом деле,

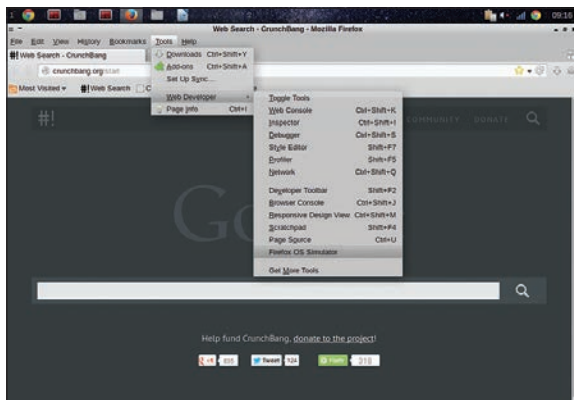
и писать приложения для Firefox OS на удивление просто, потому что в основе приложений лежат три основных открытых языка:

» **HTML5** Hyper Text Markup Language 5 — язык разметки гипертекста 5; с его помощью мы будем создавать структуру приложения.

» **CSS** Cascading style sheets — каскадные таблицы стилей; они позволяют быстро менять внешний вид приложения.

» **JavaScript** для интерактивности; в этом примере мы воспользуемся jQuery Mobile, который специально предназначен для разработки мобильных приложений.

Если у вас есть базовые знания в этих языках, то все, что осталось — отправиться в путь и начать писать приложения для Firefox OS; вам понадобится только текстовый редактор. Можете воспользоваться своим любимым — специализированных программ здесь не нужно, в отличие от других мобильных ОС и их собственных интегрированных сред разработки (IDE).



» Симулятор можно найти в меню Tools > Web Developer [Инструменты > Web-разработка].

Типовое приложение

В Firefox OS используются те же основные web-технологии, которыми мы пользуемся уже более десяти лет, и это здоровая, надежная платформа для разработки. Структура приложений обычно напоминает web-сайт, например:

manifest.webapp
index.htm
[app-icons]
...Здесь размещаются все иконки.
[scripts]
...Здесь размещается любой код на Javascript или jQuery.
[styles]
...Здесь размещаются каскадные таблицы стилей (CSS).

jQuery Mobile

jQuery Mobile, которым мы пользуемся для создания программы — замечательный язык, и на него стоит потратить время. В JavaScript, хотя и довольно простом в изучении, требуется самим разрабатывать методы создания интерактивного контента, а в jQuery есть библиотека кода, с которой очень быстро создаются контент и интерактивные компоненты для сайта или приложения.

Лучшее в jQuery то, что для создания отличных сайтов и приложений не нужно хорошо знать JavaScript. На самом деле, если вам нужна только

базовая функциональность, на JavaScript писать вообще ничего не придется — достаточно вызывать соответствующие функции в HTML. Для jQuery Mobile есть много плагинов, и их количество растет — с плагинами для галереи и даже плагином для автозаполнения формы — что делает jQuery очень мощным инструментом.

Основные ресурсы, с которыми следует ознакомиться — <http://jquerymobile.com> с информацией по сугубо мобильным ресурсам и документация на <http://jquerymobile.com/demos/1.2.1/index.html>.

Чтобы побольше узнать о jQuery, зайдите на сайт разработчиков: <http://jquery.com>.

У разработчиков, кроме того, есть превосходный онлайн-ресурс для обучения — <http://learn.jquery.com>.

jQuery Mobile также использовался для создания «родных» приложений для iOS, Android и Windows, и на его основе работает приложение PhoneGap (<http://phonegap.com>), использующее NodeJs и jQuery для быстрой разработки кроссплатформенных приложений.

Вы видите `index.htm`; здесь он играет такую же роль, какую и на обычном сайте, а другие файлы — изображения, иконки и код JavaScript — можно хранить в отдельных каталогах. Вы видите здесь и новый файл `manifest.webapp`. Этот файл используется во всех приложениях Firefox OS, и в самом простом виде он всего лишь сообщает Firefox OS, кто написал приложение и для какой цели.

Manifest.webapp — очень важный файл. Он понадобится для создания приложений и их отправки в Firefox OS Marketplace. Вот пример файла `manifest.webapp`, который я создал для нашего проекта:

```
{ "version": "0.1", "name": "Linux Command Line",
  "description": "Крутое приложение для современных линуксоидов",
  "launch_path": "index.html",
  "icons": { "16": "/app-icons/icon-16.png",
            "48": "/app-icons/icon-48.png",
            "128": "/app-icons/icon-128.png" },
  "developer": { "name": "Les Pounder",
                "url": "http://bigl.es" }
```

По сути, это файл формата JSON, который объясняет, где по отношению к нему находятся компоненты вашего сайта, и этой информацией пользуется Firefox OS для поиска компонентов. Структура манифеста довольно проста: самые важные разделы — `version` [версия], `name` [название], `description` [описание] и `launch_path` [путь запуска приложения]. Все остальное, как говорится, «неплохо бы иметь», но его отсутствие не мешает запуску приложения.

Я бы, между прочим, посоветовал включить в манифест раздел с иконками (`icons`), чтобы у приложения была своя иконка наряду с другими программами в вашем устройстве. Файл `manifest.webapp` должен быть у каждого приложения, так как это файл «аудита», который контролирует все, что касается вашего приложения.

Наше приложение

Нашим приложением станет справка по популярным командам Linux, разбитым на группы, такие как Basics [Основные], Networking [Сеть] и Raspberry Pi. Чтобы все было компактно и пользователи могли отфильтровать ненужное содержимое, воспользуемся выпадающими меню.

Область применения программы безгранична, и вы можете добавлять любой контент, который сочтете нужным, но для целей настоящего руководства я ограничусь созданием базового приложения.

Начнем с HTML-кода. Этот код создаст каркас нашего приложения, на который затем можно наложить CSS и jQuery. В следующем коде достаточно новых элементов, и я объясню их далее. Это сокращенная версия кода — полный код можно найти на **LXFDVD** и Github по ссылке https://github.com/lesp/Linux_Cheatsheet:

```
<body> <div data-role="page" id="page1">
  С помощью ролей данных (data-role) в jQuery определяется назначение компонентов, и в данном случае это страница с идентификатором page1, чтобы на нее легко было сослаться:
  <div data-theme="a" data-role="header">
    Темами данных (data-theme) управляется CSS, подключенный к jQuery; в данном случае, "a" — тема по умолчанию, и она очень похожа на Android/iOS. Атрибут data-role в данном случае создает заголовок нашего приложения, который используется как заголовок окна.
  <h3>
    Linux Cheat Sheet
  </h3>
  <div style=" text-align:center">
    
</div>
<div data-role="collapsible-set">
  <div data-role="collapsible" data-collapsed="true">
    Здесь начинается выпадающий div, который можно сделать открытым (true) или закрытым (false) по умолчанию.
  <h3>
    Команды
  </h3>
  <div data-role="collapsible" data-collapsed="true">
    Я добавил еще один выпадающий div, чтобы аккуратнее и более ступенчато организовать вложенный контент: так наша страница будет чище и аккуратнее.
  <h3>cd</h3>
  <p>Меняет текущую директорию на другую, позволяя перемещаться по файловой системе.</p>
  </div>
  Теперь перейдем к нижней части страницы:
  <div data-theme="a" data-role="footer" dataposition="fixed">
    Это нижняя часть приложения, и в ней я вывел название приложения:
  <h3>
    Linux Cheat Sheet
  </h3>
  </div>
  </div>
  </div>
  </div>
  </div>
  </div>
  </div>
  </body>
```

»

jQuery Mobile — прекрасная библиотека JavaScript, она очень проста в использовании и специально предназначена для экранов мобильных телефонов и планшетов. Проводя исследование для этой статьи, я пересмотрел разные мобильные библиотеки для JavaScript, и jQuery Mobile вышла безусловным победителем. В ней есть множество компонентов HTML, таких как диалоговые окна, выпадающие меню и плавные переходы, которые легко поместить в приложение. Все они будут прекрасно смотреться в вашей программе.

jQuery Mobile создавалась теми же, кто работал над jQuery, так что у нее хорошая родословная, и jQuery Mobile пользуются многие известные компании, такие как IKEA и Disney.

Включение jQuery Mobile

Обычно мы сообщаем приложению, где установлена библиотека jQuery. Это можно сделать двумя способами: через локальную копию библиотеки jQuery Mobile, которая связывается с приложением и устанавливается на телефон или планшет пользователя, или через удаленную копию, размещенную в сети доставки контента (CDN, Content Delivery Network), такой как jQuery или Google.

Второй вариант представляет собой прекрасный способ защиты будущего ваших разработок, так как любые изменения в библиотеках сразу же доступны приложению. В нашем проекте мы воспользуемся онлайн-версией jQuery, расположенной на удаленной CDN. Это означает, что нужно позаботиться о ссылке на CDN где-то в коде. Вот код, который обрабатывает размещение библиотек jQuery и файлов CSS:

```
<link rel="stylesheet" href="https://d10ajooocyu32n.cloudfront.net/mobile/1.3.1/jquery.mobile-1.3.1.min.css">
<!-- Extra Codiqa features -->
<link rel="stylesheet" href="codiqa.ext.css">
<!-- jQuery and jQuery Mobile -->
<script src="https://d10ajooocyu32n.cloudfront.net/jquery-1.9.1.min.js"></script>
<script src="https://d10ajooocyu32n.cloudfront.net/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>
<!-- Extra Codiqa features -->
<script src="https://d10ajooocyu32n.cloudfront.net/codiqa.ext.js"></script>
</head>
```

Использование тем данных jQuery

В составе jQuery Mobile есть собственный шаблон стилей CSS, и используются различные темы для изменения внешнего вида приложения. Чтобы изменить тему, можно изменить букву

в атрибуте `data-theme="a"` на другую (они могут принимать значения от **a** до **e**).

Впрочем, старайтесь придерживаться постоянства в выборе тем: это приведет к более профессиональному результату. Тема данных **a** — темная, с белым текстом. Тема данных **b** напоминает раскраску Twitter, в ней много синего и серого. В теме данных **c** присутствуют серые и белые цвета, **d** — темная версия темы **c** и, наконец, **e** — яркая, пикантно-желтая тема.

Для нашего проекта мы воспользуемся встроенным CSS, который есть в jQuery Mobile. Я выбрал `data-theme="a"` — она отлично смотрится. Но вы вольны редактировать файлы CSS согласно потребностям приложения.

Mozilla выпустила отличное дополнение-плагин к Firefox, предоставляющее полноценную среду тестирования Firefox OS; оно называется Firefox OS Simulator и пока что в альфа-версии, но это очень надежный и удобный способ протестировать приложения в среде, похожей на реальные устройства. Для установки дополнения зайдите в Tools > Add-Ons [Инструменты > Дополнения] и наберите в поиске "Firefox OS Simulator". Когда плагин будет обнаружен, установите его. После установки вы найдете симулятор в меню Tools > Web Developer [Инструменты > Web-разработка].

Тестирование приложения

При первом запуске симулятора вы увидите, что несколько примеров приложений там уже есть, и их легко можно запустить, нажав кнопку на левой панели. Они запустятся, и вы увидите главное окно Firefox OS.

Для просмотра установленных программ шаркните по экрану мышью справа налево. Должна появиться знакомая сетка из ярлычков, и в нижнем ряду должны быть примеры приложений. По щелчку на иконке левой кнопкой мыши запустится демонстрационное приложение.

Для установки вашего приложения нажмите Add Directory [Добавить каталог]; откроется диалоговое окно, где нужно выбрать файл `manifest.webapp` своего приложения. После выбора манифеста симулятор автоматически запустит приложение.

Также можно протестировать свое приложение на реальном устройстве, например, на ZTE Open [Обзоры, стр. 16, LXF177], и это можно сделать в симуляторе Firefox OS, подключив свой телефон. Подробная информация приведена во врезке внизу. В Firefox OS есть два типа приложений:

» **Hosted** [Размещенное на сервере] В этом случае приложение размещается на сервере в Интернете. Для статических сайтов (только на HTML/CSS и без PHP и т. п.) Mozilla рекомендует

Советы и инструменты для разработчика

Разработка приложений для Firefox OS довольно проста, и Mozilla создала замечательный ресурс, который помогает создавать отличные приложения. Mozilla Developer Network теперь включает Firefox Marketplace Developer Hub с руководствами по созданию, компиляции и публикации приложений. Чтобы зайти на ресурс, введите в адресной строке браузера <https://marketplace.firefox.com/developers> и следуйте превосходному пошаговому руководству.

На момент написания заметки последней версией Firefox OS все еще оставалась 1.0.1.0, и дополнительную информацию о ней и о сопутствующем API можно получить на www.mozilla.org/en-US/firefox/os/notes/1.0.1.

Симулятор Firefox OS — прекрасная утилита, с помощью которой можно быстро протестировать приложение в виртуальной среде Firefox OS. Также можно связать с Firefox OS телефон (такой, как упомянутый нами ZTE Open, о котором мы писали в LXF177 [Обзоры, стр. 16]), чтобы разработчик мог отправлять свои приложения для тестирования прямо на свои устройства, в обход Marketplace.

Настроить такое немного сложно, но в Linux это чуть проще, потому что нужно лишь изменить конфигурационный файл. Также нужно включить в телефоне удаленную отладку (она находится в меню Settings > Device Information > More Information > Developer [Настройки > Информация об устройстве > Подробнее > Разработка]) и затем

изменить файл `/etc/udev/rules.d/51-android.rules` (или создать его, если его еще нет), чтобы он содержал следующую строку:

```
SUBSYSTEM=="usb", ATTR{idVendor}=="19d2",
MODE="0666", GROUP="plugdev"
```

Кстати, 19d2 — идентификатор производителя для телефона ZTE Open; для устройств Geeksphone нужно использовать 05c6.

Подробнее об этом можно узнать на прекрасных интернет-ресурсах Mozilla: https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS/Debugging/Connecting_a_Firefox_OS_device_to_the_desktop, а более подробно об Android Debug Bridge (adb) — в руководстве по Android: <http://developer.android.com/tools/device.html>.

Github Pages (<http://pages.github.com>). Если ваше приложение динамическое, т.е. в нем используется PHP/Python или другой язык программирования, лучше разместить его на сервере, и есть множество компаний, предоставляющих свои серверы с разными версиями Linux.

» **Packaged** [В пакете] Приложение в пакете — открытое приложение, все ресурсы которого (HTML, CSS, JavaScript, манифест приложения и т.д.) содержатся в ZIP-архиве, а не на web-сервере. Приложение в пакете — это просто ZIP-архив с файлом манифеста в корневом каталоге. Файл манифеста должен называться **manifest.webapp**.

Отличие от размещаемого на сервере приложения в том, что в манифесте приложения в пакете нужно указать `launch_path` [путь запуска], тогда как для размещаемого на сервере приложения это необязательное поле.

В пакете имеется три типа приложений:

» **Privileged** [Привилегированное] Приложение, одобренное Firefox Marketplace; оно проходит специальный процесс проверки, чтобы его безопасность для пользователей была гарантирована. Обычно такое приложение также пользуется специальным API на планшете или телефоне, на который устанавливается.

» **Certified** [Сертифицированное] Предназначено для критически важных задач, таких как совершение вызова. Такие приложения не могут применяться сторонними разработчиками и требуют серьезной сертификации от производителей и поставщиков оборудования.

» **Plain** [Обычное] Самое простое, не требующее особой проверки или сертификации. При загрузке приложения на Marketplace оно подписывается Marketplace, после чего его могут загружать другие пользователи.

Для нашего примера мы выбрали последний вариант — обычное приложение.

Отправка вашего приложения

Процесс отправки приложения тоже относительно прост, и есть лишь несколько моментов, которые нужно принять во внимание. Во-первых, зарегистрируйтесь на сайте с учетной записью Persona. Persona — собственный метод аутентификации Mozilla, он похож на openID.

Для первого приложения советую выбрать вариант с размещением на сервере, и отправить приложение на Firefox OS Marketplace. Подумайте, хотите ли вы, чтобы приложение было доступно на Marketplace сразу после одобрения. По умолчанию происходит именно так. Если вам этого не надо, уберите галочку `Publish my app in the Firefox Marketplace as soon as it's reviewed` [Опубликовать мое приложение в Firefox Marketplace сразу после одобрения] при отправке приложения. Затем понадобится сделать следующее:

» Для приложения с размещением на сервере нужно указать URL файла манифеста вашего приложения.

» Вам понадобится web-хостинг для приложения (но для приложений в пакете он не нужен). Если вы выбрали приложение с пакетом, поместите его в ZIP-файл с манифестом в корневом каталоге.

» Не забудьте создать как минимум одну иконку размером не менее 128×128 пикселей.

» Также нужно создать как минимум один экранный снимок приложения.

» Не забудьте указать правила использования программы или URL страницы с ними.

» Укажите e-mail поддержки для своего приложения.

» Убедитесь, что вы поняли критерии обзора приложения Marketplace.

Выполнив эти требования, вы можете последовать указанию руководства на <https://marketplace.firefox.com/developers/submit> и отправить свое первое приложение на Marketplace. Для первого приложения я бы посоветовал выбрать тип Plain (как я объяснял ранее): такие приложения проверяются быстрее всего, и, следовательно, ваше приложение скоро будет доступно. Я бы также посоветовал разместить приложение на собственном web-сервере и предоставить Firefox OS Marketplace URL файла **manifest.webapp** и любую другую необходимую информацию.

На появление приложения в Firefox OS Marketplace уйдет некоторое время, так как перед одобрением Mozilla должна провести осмотр приложения и провести контроль качества; но после одобрения вы получите соответствующее сообщение от Mozilla. После этого вы сможете пользоваться им в Firefox OS.

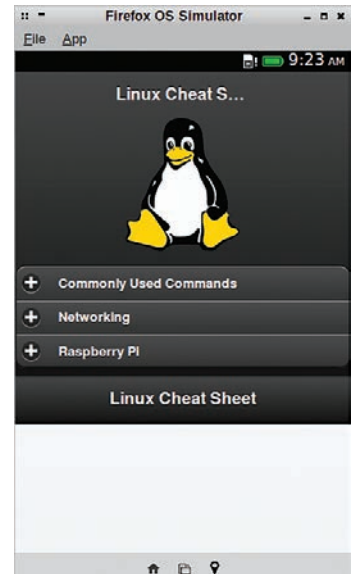
Идем дальше

Так как в основе Firefox OS лежат открытые технологии, есть множество API, которые можно исследовать и опробовать: от простых действий (когда вы выбираете, что делать с файлом, таким как электронное письмо, галерея или твит) до сложных, которые управляют светом, скоростью и географическим положением.

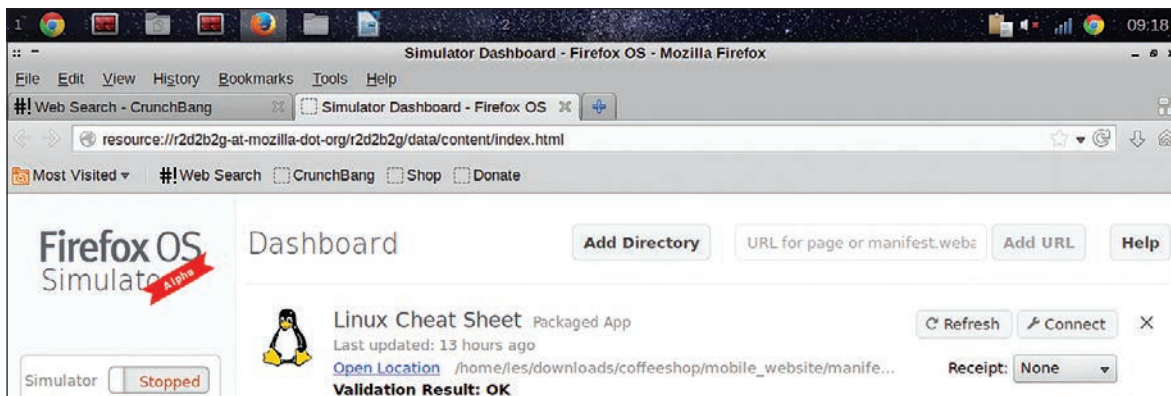
Экспериментируя с этими API, можно создавать полноценные приложения только из HTML, CSS и JavaScript — других языков не требуется. Если эти приложения размещаются на web-серверах, легко создать одно приложение, которое будет хорошо работать на различных мобильных платформах.

На этом наш проект закончен. Мы выложили на Firefox Marketplace базовое приложение, готовое для загрузки пользователями по всему миру. Но не останавливайтесь на этом. Создание приложений для Firefox OS будет способствовать ее развитию, а с обширным набором свободного API — весь мир у ваших ног.

Пишите приложения, и они придут. Не призраки мертвых бейсболистов, конечно, а пользователи. **LXF**



» Наше приложение *Cheatsheet*, запущенное в Firefox OS Simulator.



» Главный экран Firefox OS, который можно увидеть на многих устройствах.

Сравнение

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

Linux — новичкам

Созрели рвануть в Linux, но не уверены, с какого дистрибутива начать? Шашанк Шарма перечисляет пять лучших вариантов для начинающих.



Про наш тест...

Все дистрибутивы тестировались на одной и той же двудюймовой машине с 4 ГБ ОЗУ. Мы брали свежие стабильные релизы всех дистрибутивов, кроме SolusOS: он очень сильно изменился со времени своего последнего стабильного релиза, поэтому для нашего Сравнения мы выбрали альфа-релиз.

Для неопытных пользователей документация является одной из главнейших причин предпочесть дистрибутив. Дистрибутив также должен быть прост в установке. Поскольку большинство новообращенных, вероятно, никогда еще не устанавливали Linux, это очень важная функция. Не менее важны управление программами и приложениями, включенные в дистрибутив.

Помимо этого, дистрибутив должен быть прост в повседневном использовании. Идеальный дистрибутив для начинающих — тот, который объединяет в себе все вышеперечисленное, а также облегчает им возможность сделать кое-какие мелкие настройки.

Наша подборка

- » Elementary OS
- » Pear Linux
- » PinguyOS
- » SolusOS
- » Zorin OS

Изобилие выбора — одна из главных проблем начинающих линуксоидов. Выбор вашего первого Linux способен вызвать растерянность. И особенно — если вы даже не знаете, какими критериями руководствоваться при выборе дистрибутива.

В середине и конце 1990-х процесс выбора дистрибутива был куда проще. Вы хватались за дистрибутив, о котором слышали или с которым кто-то из ваших знакомых работал, или за тот, у которого была хоть какая-то документация. Естественно, ваш выбор ограничивался Red Hat, Debian или Slackware. Те же критерии срабатывают и теперь, но разрастание

«В этом Сравнении мы намеренно избегали самых популярных дистрибутивов.»

количества дистрибутивов Linux и их громкие сообщества затрудняют возможность осесть на одном из них и приступить к работе.

Мы намеренно избегали самых популярных дистрибутивов, поскольку нам не были нужны дистрибутивы, характерные только простотой в использовании. Вместо этого мы выбрали пять тех, которые, по нашему мнению, являются идеальными стартовыми точками.

Ubuntu очень давно является популярным дистрибутивом Linux, однако он не вполне подходящий для начинающих. Хотя и может стать таковым при условии внесения некоторых изменений. Именно поэтому четыре дистрибутива в нашем списке основаны на Ubuntu.

Для тех пользователей, которые хотят найти знакомые черты, отказавшись от проприетарной ОС, у нас есть дистрибутивы, напоминающие Mac и Windows.

Включенные программы

Что они предлагают сразу после установки?

Обычно дистрибутивы разрабатываются так, чтобы обслужить как можно больше потенциальных пользователей. Та же философия лежит в основе создания включенных в них приложений. Все дистрибутивы в нашем списке предлагают минимум: интернет-браузер, клиент электронной почты, текстовый редактор и мультимедиа-плеер. Но если вы рассчитываете найти побольше приложений, такое тоже есть!

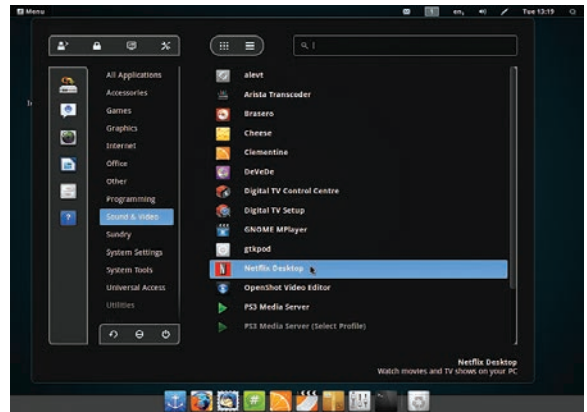
Текущий релиз SolusOS предназначен для разработчиков и тестировщиков, и в нем ограниченное число приложений. Стоит отметить *Firefox 24.0b9* и *Thunderbird 17.0.8*. Разработчик уже объявил о своих планах включить клиент Steam в финальный релиз, и можно рассчитывать на все виды производственных и мультимедиа-приложений.

Zorin просто набит приложениями. Вы получаете обычные программы для офиса и для Интернета, такие, как *LibreOffice* и проприетарный *Google Chrome*. Также дистрибутив позволяет просматривать контент в проприетарных форматах из среды live. В него включены редактор изображений *GIMP*, менеджер фотографий *Shotwell*, *Thunderbird*, *Pidgin* для IM,

видеоплеер *Totem*, музыкальный плеер *Rhythmbox* и видеоредактор *OpenShot*. Есть еще *Wine* и *PlayOnLinux* для установки игр и приложений, работающих только в Windows. Здесь также есть *Web Browser Manager*, который облегчает установку разных браузеров. Дистрибутив содержит *Gwibber* — приложение рабочего стола, позволяющее управлять большинством популярных социальных сетей, таких, как Twitter, Facebook, Google+ и Flickr.

Elementary OS отличается эстетичным, элегантным дизайном и подбором приложений с простой структурой. Это объясняет включение клиента электронной почты *Geary*. Большинство других дистрибутивов идут с *Thunderbird*, хотя *Geary* — тоже хороший клиент электронной почты. Также у вас есть менеджер фотографий *Shotwell*, видеоплеер *Totem*, музыкальный плеер *Noise* и веб-браузер *Midori*. Два последних еще раз подтверждают приверженность дистрибутива к легковесным, простым приложениям. Elementary предлагает меньше пакетов по умолчанию, и вам понадобится его менеджер приложений для установки ПО, которого вам не хватает.

Pear OS идет с *LibreOffice*, *Thunderbird*, *Firefox 20*, мультимедиа-плеером *VLC*



➤ Наведя мышью на приложение, вы увидите, что делает каждое из них.

и музыкальным плеером *On Air*. Кроме того, здесь имеются *Time Back*, клон инструмента резервного копирования *Time Machine* от Apple, *Empathy* для IM и инструмент веб-камеры *Cheese*.

В составе PinguyOS — *Thunderbird*, *LibreOffice*, *Empathy*, *Deluge*, *TeamViewer 7*, *DeVeDe* для записи дисков, редактор видео *OpenShot*, *Cheese*, *Clementine*, *Gparted 0.16.1*, *Shutter*, *PlayOnLinux* и много чего еще. Включение *TeamViewer* выглядит ловким ходом, поскольку приложение позволяет управлять удаленным рабочим столом.

Вердикт

- PinguyOS ★★★★★
- Zorin OS ★★★★★
- Elementary OS ★★★★★
- Pear Linux ★★★★★
- SolusOS ★★★★★

» Pinguy предлагает несколько неортодоксальных, но блестящих пакетов.

Управление программами

Для случаев, когда надо установить добавочные приложения.

Большинству новичков набора приложений по умолчанию должно быть более чем достаточно для начала. Но по мере привыкания к своему дистрибутиву вам может захотеться установить дополнительные приложения.

Программные репозитории могут сперва показаться очень странной концепцией, однако большинство дистрибутивов предлагают полезные и удобные инструменты для несложной установки программ.

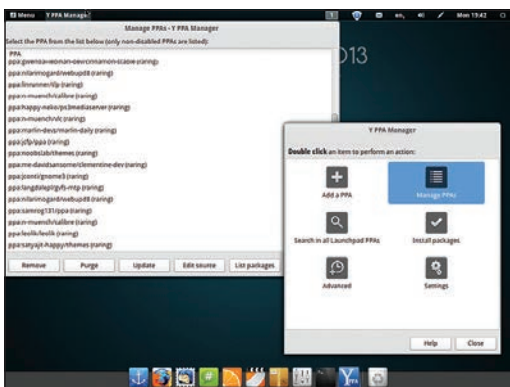
SolusOS использует в качестве интерфейса к установке приложений *gpk-application*, но разработчик трудится над индивидуальным GUI для своей системы управления пакетами *PiSi*.

Elementary OS работает на собственном AppCenter, заменившем Ubuntu Software Center. За этим исключением, все остальные дистрибутивы на базе Ubuntu в нашем Сравнении идут с *Ubuntu Software Center* и *Synaptic Package Manager*.

В Pear Linux ни один из этих инструментов не доступен с рабочего стола. Сначала вы должны перейти в Launcher, а затем щелкнуть по вкладке System Tools [Системные инструменты]. По умолчанию включены репозитории Main [основной], restricted [ограниченный] и multiverse [мультиверсионный].

В Zorin вы получите переработанный *Ubuntu Software Center*, а также *Synaptic Package Manager*. Помимо обычных, здесь есть репозитории для Google и Opera.

PinguyOS имеет ряд дополнительных программных репозиториях, включенных по умолчанию, а заодно прихватил репозитории Linux Mint, Ubuntu и Elementary OS. Имеются PPA для тем и приложений наподобие *Clementine*. Дистрибутив поставляется с огромным количеством PPA, но, к счастью, у него есть *Y PPA Manager*. Это инструмент, с помощью которого вы освоите все PPA и управитесь с ними.



➤ Ух ты, сколько PPA!

Вердикт

- Pear Linux ★★★★★
- PinguyOS ★★★★★
- Zorin OS ★★★★★
- Elementary OS ★★★★★
- SolusOS ★★★★★

» SolusOS проигрывает, поскольку управление пакетами — дело важное.

Удобство использования

Легко ли будет новичкам настроить рабочий стол?

Можно сказать, что пользователь нашел свой дистрибутив, когда он начинает отходить от настроек по умолчанию — признак зрелости любого пользователя, но вернее всего это для новоиспеченных линуксоидов.

Часто говорят, что дистрибутивы Linux исключительно настраиваемые. Но что это означает для

новичков? Ну да, можно изменить фон рабочего стола, тему значков, определить горячие клавиши, настроить управление питанием и внести другие изменения во внешний вид и поведение дистрибутива. Но легко ли новичку в Linux сделать все это?

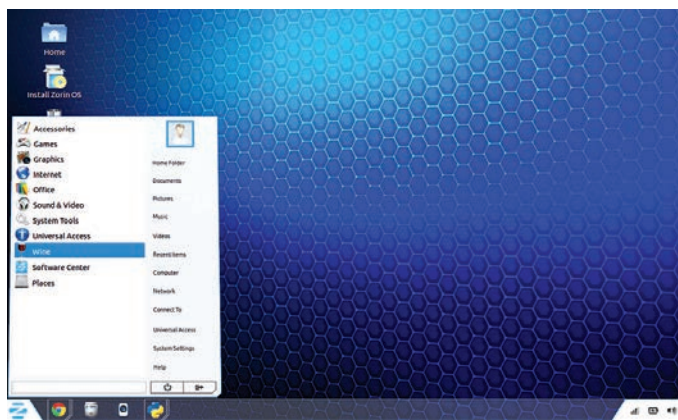
Хотя все дистрибутивы в нашем списке позволяют делать все это и даже больше, каждый

из них подходит к этому процессу по-своему. Если дистрибутив предназначен для начинающих, он заработает много очков, предусмотрев специальный инструмент настройки, способный помочь неопытным пользователям настроить все аспекты дистрибутива согласно собственным предпочтениям.

SolusOS ★★★★★

Основанный изначально на Debian, сейчас SolusOS разрабатывается с нуля разработчиком, который работал над Debian-редакцией Linux Mint. Хотя дистрибутив находится на стадии активной разработки и еще не дошел до бета-релиза, он весьма стабилен.

Программа установки релиза Alpha 9 примитивна и даже не имеет инструмента для разбиения диска на разделы, однако разработчик заимствует этот инструмент из YALI, программы установки Pardus. Финальная SolusOS 2 будет идти только с устанавливаемыми образами, без опции live. Дистрибутив использует собственную адаптацию менеджера пакетов PiSi, созданную для Pardus Linux. Поддерживаются дельта-обновления, позволяющие держать систему в актуальном состоянии при минимальной пропускной способности канала. SolusOS использует Xfce, однако полный дистрибутив будет выходить с Consort, собственным ответвлением рабочего стола Gnome Classic.



Zorin OS ★★★★★

Это один из лучших дистрибутивов для привлечения неопытных пользователей Linux. В нем есть все, чтобы предложить отличную работу пользователю, пришедшему с другого дистрибутива Linux или даже с Windows или Mac OS X.

Помимо рабочего стола в стиле Windows 7, индивидуально настраиваемая программа запуска приложений отлично справляется с имитацией меню Пуск Windows 7. Редакция Core предлагает достаточно, чтобы раззадорить ваш аппетит, а потратив немного денег, вы получите специализированные версии.

Zorin включает все прелести Ubuntu, такие, как *Ubuntu One*, хорошо интегрированный в дистрибутив. Также он привлекает пользователей неплохой навыв работы, периодически напоминая им о необходимости настроить приложение резервного копирования. В общем и целом, дистрибутив предлагает нужную смесь лучшего от Ubuntu и некоторых спецприложений Zorin, таких, как *Look Changer*.

Коммерческие сервисы

Какие платные дополнения предлагает дистрибутив?

У дистрибутива может наличествовать несколько причин предлагать платные дополнения. Чаще всего разработчики просто хотят добыть денег на продолжение своей деятельности. Именно поэтому некоторые дистрибутивы также облегчают пользователям способы пожертвовать на проект.

В дополнение к релизу рабочего стола, Zorin OS производит четыре превосходных версии, которые можно загрузить, сделав пожертвование. Версии Business, Multimedia и Gaming можно получить за мини-

мальное пожертвование в размере €7,99, а редакцию Ultimate — за пожертвование не менее €9,99. С приобретением этих изданий вы также получаете отличную поддержку. Дистрибутив к тому же продает сувенирную продукцию — кружки, футболки и стикеры — на CafePress.

PinguyOS также предлагает широкий ассортимент на CafePress, где можно выбрать разную сувенирную продукцию, например, кружки, футболки, сумки или детские слюнявочки. Пожертвование делается через PayPal.

Чтобы выразить свою поддержку Pear Linux, можно приобрести стикеры. Или сделать пожертвование; опций несколько, от минимума €5 до максимума €100.

Можно еще стать спонсором, обеспечив выдачу разработчикам зарплаты. Уровней спонсорства пять: Platinum, Gold, Silver, Bronze и Community. Спонсоры платят фиксированную сумму ежемесячно, что отображается на сайте и в ежемесячных уведомлениях. Магазин Elementary продает CD с текущим 32-битным релизом, а также предлагает футболки и стикеры.

Вердикт

- Pear Linux ★★★★★
- PinguyOS ★★★★★
- Zorin OS ★★★★★
- Elementary OS ★★★★★
- SolusOS ★★★★★

» SolusOS не имеет коммерческих сервисов, но это дистрибутив не бедный.

Elementary OS ★★★★★

Это один из простейших дистрибутивов на базе Ubuntu, а также отличный стартовый пункт для начинающих.

Дистрибутив делает явный упор на дизайн, что привело к весьма любопытному выбору пакетов программ по умолчанию. Хотя эти программы могут и не всем прийти по вкусу, они тем не менее очень удобны и отлично заменяют свои более популярные альтернативы.

Дистрибутив легок и отличается невероятной быстротой. «Из коробки» он не предлагает вам сверхбогатого выбора приложений, и не включает кодеков для проприетарных медиа-форматов. Это означает, что воспроизвести MP3, видео или даже видео с YouTube сразу после установки у вас не получится; но вы можете воспользоваться наследием Ubuntu и получить доступ к тысячам дополнительных пакетов и медиа-кодеков через Центр приложений.



Pear Linux ★★★★★

Этот дистрибутив стремится внешне походить на Mac. Подобно Elementary, он не предлагает кучи приложений сразу же, что довольно редко для приложений на основе Ubuntu. Сейчас он доступен только как 64-битный ISO, но в будущих релизах будет и 32-битный вариант, начиная с Pear OS 8.

Помимо настольной редакции Desktop, команда производит редакцию Server, с Apache, MySQL, Samba, Webmin, TomCat и т. д., пригодную для бизнеса, домашнего офиса, школы. Также она включает спецприложение My-Server, в помощь управлению разными сервисами. В настольной есть MyPear, универсальный инструмент для управления и настройки многих аспектов рабочего стола: например, определения горячих углов, размещения панелей, анимации окон и многого другого. Имеется здесь и Clean My Pear, способствующий работе с системой. Вы можете использовать этот инструмент, чтобы избавляться от мусора и удалять временные файлы и кэш браузера.

PinguyOS ★★★★★

Традиционно любое приложение или операционная система, имеющая в своем названии слово Бета, подразумевает, что проект не готов для массового производства, однако Pinguy перевернул эту теорию с ног на голову, и все его выходящие каждые полгода финальные релизы именованы Бета.

Дистрибутив удивительно стабилен и является очень привлекательной опцией для всех пользователей Linux. Являетесь ли вы абсолютным новичком или просто хотите перейти на другой дистрибутив, он определенно стоит вашего времени.

В нем имеется специальная утилита Docky, используемая для создания ряда доков. В каждый такой док вы можете добавлять доклет [docklet]: например, погода, монитор использования сети и переключатель рабочих областей. Также дистрибутив включает инструмент Tweak, с которым вы легко настроите много разных аспектов рабочего стола.



Поддержка и документация

RTFM не помогает. Что еще у вас есть?

Независимо от прошлых шапней с другими ОС, новичок в Linux столкнется с совершенно иным принципом работы; ему придется привыкать ко всему, от внешнего вида и до альтернативных приложений.

Именно поэтому дистрибутив должен предлагать обширную документацию. Дополнительные ресурсы, такие как форумы, списки рассылки, wiki и т. д., которые способны помочь новичку воспользоваться накопленным опытом сообщества, также приветствуются.

Elementary OS поместил точную и понятную документацию на сайте; проект имеет страничку Answers, где каждый может задать вопрос. А к IRC-каналу проекта можно подключиться прямо из браузера.

У SolusOS — хорошо организованный форум, где размещены разработанные сообществом руководства, советы, подсказки и хитрости.

Zorin OS предлагает только минималистский справочник по установке, но более чем справляется с задачей, при множестве своих FAQ для новых пользователей, фо-

румов с помощью по установке, HowTo и руководств. Также у проекта есть IRC-канал, и вы можете немедленно получить ответы на свои вопросы.

PinguyOS предлагает своим пользователям то же, что и Zorin, кроме FAQ. Ко всему хорошему, у него есть пошаговое руководство по установке.

В противоположность всем этим дистрибутивам, Pear Linux предлагает форумы и мало что сверх того. И хотя сайт извещает о wiki, на данный момент она недоступна.

Вердикт

- PinguyOS ★★★★★
- Zorin OS ★★★★★
- Elementary OS ★★★★★
- SolusOS ★★★★★
- Pear Linux ★★☆☆☆

» Документация может стать одной из самых важных функций.

Цикл релизов

Какое расписание оптимально для начинающего?

Есть три популярных методологии разработки, которых обычно придерживаются дистрибутивы Linux: жесткое расписание, фиксированные функции и возобновляемый релиз. Возобновляемый релиз предполагает некоторую сложность в освоении для новичков. Именно по этой причине дистрибутивы вроде Gentoo и Arch

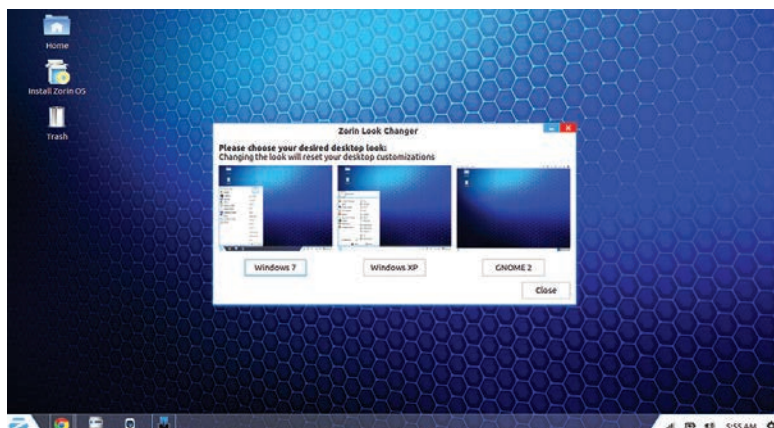
начинающим не рекомендуются. С фиксированными функциями, дистрибутив выходит, когда он готов и достиг определенного качества — здесь не устанавливается дата выпуска. SolusOS — единственный дистрибутив в нашем списке, который следует данной методике. Он создается полностью с нуля, поэтому у него нет «вышележащего [upstream]» источника.

Жесткое расписание — один из самых популярных циклов релизов, и ему следует большинство дистрибутивов. С жестким расписанием новый релиз выходит через определенные промежутки времени, обычно раз в полгода. Циклу, предполагающему выход нового релиза дважды в год, следует Ubuntu — и, естественно, его производные ему вторят.

Zorin OS основан на самом свежем релизе Ubuntu. Работа над новой редакцией начинается, как только на горизонте появляется новый релиз Ubuntu, однако разработчику требуется время на создание разных редакций.

Текущий релиз Elementary OS основан на Ubuntu 12.04 LTS. Однако он не будет производить дистрибутив на основе Ubuntu 13.10. Следующий релиз Elementary, под названием Isis, возьмет за основу Ubuntu 14.04.

Выходящие раз в полгода релизы PinguyOS идут с самыми современными программами и не считаются стабильными. Они остаются в бета-версии, хотя и являются окончательным релизом. Стабильные релизы основаны на релизах Ubuntu LTS.



» Zorin выходит с несколькими специнструментами, типа *Look Changer*, который в свободно скачиваемой редакции предлагает на выбор внешность Windows 7, XP или Gnome 2.

Вердикт

- Elementary OS ★★★★★
- Pear Linux ★★★★★
- PinguyOS ★★★★★
- SolusOS ★★★★★
- Zorin OS ★★★★★

» Дистрибутивы могут быть основаны на Ubuntu, но по ним этого не скажешь.

Установка

Насколько дружелюбен к пользователю этот процесс?

Live CD — лучшее, что Linux может предложить новым пользователям. Возможность протестировать дистрибутив и познакомиться с его внешним видом и содержанием, не устанавливая его, помогает новичкам освоить Linux.

Все дистрибутивы здесь можно тестировать из среды Live. Наигравшись с дистрибутивом и ощутив, что он вам подошел, запускайте его установку прямо из среды Live. У большинства этих дистрибутивов есть на рабочем столе значок, двойной щелчок по которому запустит установку.

Раз пользователь неопытен, установка должна быть проста. Скорее всего, у пользователя уже есть какая-то ОС на компьютере, и ему придется разбивать на разделы жесткий диск. Многие популярные дистрибутивы терпят здесь неудачу, потому что не предоставляют достаточно дружелюбную программу установки. Но это не проблема дистрибутивов для новичков.

Подобно практически всем аспектам SolusOS 2, ее программа установки

находится в стадии активной разработки. Имеющаяся программа основана на той, которую разработчик написал, работая над LMDE, и позднее использовал в SolusOS 1. На своей текущей стадии программа установки пока что только базовая — в ней даже нет разбиения диска на разделы. А значит, пользователям придется использовать другие инструменты, чтобы подготовить свой жесткий диск для дистрибутива. Разработчик трудится над адаптацией YALI, инструмента разбиения диска на разделы из Pardus.

Все дистрибутивы на базе Ubuntu используют программу установки Ubuntu — *Ubiquity*, слегка подправленную, чтобы лучше соответствовать уровню новичков.

Обычно процесс установки проводит вас через семь стадий, которые включают разбиение диска на разделы, создание пользователя, определение часового пояса и раскладки клавиатуры. Самый важный шаг — разбиение диска, где вы стираете информацию со всего своего диска



» Все производные Ubuntu имеют сходные процессы установки.

и используете его для установки дистрибутива, либо указываете другие варианты разбиения. Лучшее в *Ubiquity* для новичка — большое количество документации. Плюс видео на YouTube, которые проведут вас по процессу установки для каждого из наших дистрибутивов на базе Ubuntu.

Поскольку дистрибутивы основаны на Ubuntu, установленные программы выбирать не приходится. Стоит вам только указать диск и настроить разделы, как дистрибутив автоматически установит программы за вас.

Вердикт

- Elementary OS ★★★★★
- Pear Linux ★★★★★
- PinguyOS ★★★★★
- Zorin OS ★★★★★
- SolusOS ★★★★★

» Непревзойденный в своей простоте Ubiquity получает много очков.

Дистрибутивы для новичков

Вердикт

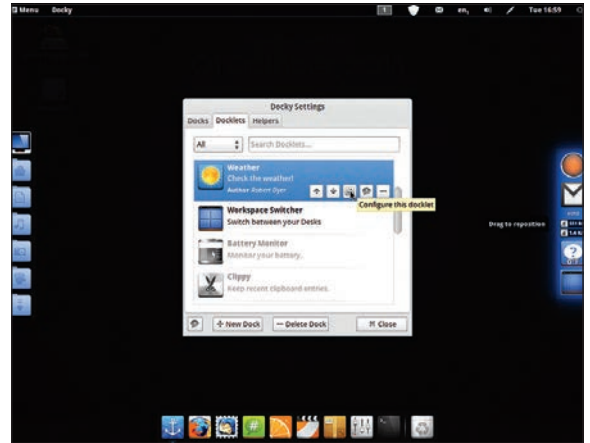
Экосистему Linux часто хвалят, а иногда и ругают, за слишком большой выбор, даваемый пользователю. И это касается не только приложений, но и дистрибутивов. До недавних лет среди опытных линуксоидов считалось крутым жаловаться на то, что дистрибутивы плодятся, как кролики, но из этого мало что вышло. Потом стали писать, что скоро мы увидим год Linux на рабочем столе. О чем говорят и до сих пор.

Однако есть среди нас и те, кого приводит в трепет известие о появлении каждого нового дистрибутива Linux. У новичков в Linux при просмотре списка дистрибутивов может закружиться голова, но это не так уж плохо. Это просто означает, что где-то обязательно есть дистрибутив именно для вас. Но не ждите же вам, пока этот скроенный специально на вас дистрибутив упадет вам в руки, чтобы начать работу в Linux! Так с чего же вам начать? Как насчет одного из наших дистрибутивов специально для новичков?

SolusOS оказался худшим в списке, судя по рейтингу. Но это многообещающий дистрибутив, к которому мы настоятельно рекомендуем обратиться, когда выйдет его финальная версия. Мы сравнивали альфа-релиз с более зрелыми, стабильными дистрибутивами, но проект все равно выделяется. SolusOS 2 сильно превосходит последний стабильный релиз, и в проекте есть отличные идеи, способные привлечь и удержать пользователя.

Pear Linux идеален для тех, кто намерен отказаться от Mac, но хотел бы видеть сходство с ним. Он не так отполирован, как Elementary, но над ним активно работают, и его специальный набор инструментов заставляет его запомнить.

Elementary OS начинал как претендент на первенство, однако из-за ряда недочетов, например, скудного набора приложений по умолчанию, пришел к финишу третьим.



Сильную конкуренцию ему составил француз Pear Linux, и, возможно, вскоре его превзойдет.

Борьба между Zorin и PinguOS была жесткой. Zorin производит несколько коммерческих вариантов и включает специнструменты. Но для начинающих линуксоидов PinguOS предоставляет лучший опыт работы. Он прост в применении и настройке, и его дизайн интригует.

» Доки и доклеты могут сделать рабочий стол привлекательным, если понять принцип их работы.

«PinguOS предоставляет лучший опыт работы. Он прост в применении, и его дизайн интригует.»

I PinguOS ★★★★★
 Сайт: www.pinguos.com Лицензия: GPL и другие Версия: 12.04
 » Приятный в использовании дистрибутив. Идеален для новичков.

IV Pear Linux ★★★★★★
 Сайт: <http://pearlinux.fr> Лицензия: GPL и другие Версия: Pear Os 7
 » Немного усилий вполне могут возвести его на пьедестал.

II Zorin OS ★★★★★★
 Сайт: www.zorin-os.com Лицензия: GPL и другие Версия: Zorin OS 7
 » Очень продуманный дистрибутив. Хорош для большинства начинающих.

V SolusOS 3 ★★★★★★
 Сайт: www.solusos.com Лицензия: GPL и другие Версия: 3
 » Подождите его вычеркивать до выхода финального релиза.

III Elementary OS ★★★★★★
 Сайт: www.elementaryos.org Лицензия: GPL и другие Версия: Luna
 » Хоть и рядом не стоит с Pingu или Zorin, но очень удобен в использовании.

Обратная связь
 Вы — начинающий линуксоид, который уже вкусил некоторые дистрибутивы? Сообщите нам о своих фаворитах по адресу lxr.letters@futurenet.co.uk.

Рассмотрите также...

Мы решили не брать популярных дистрибутивов. Кое-кто отрицает такое явление, как дистрибутивы для новичков: мол, решительно настроенный пользователь найдет способ освоиться с дистрибутивом, каким бы чуждым он ни казался.

Часто видишь, что новичкам рекомендуют Arch и Gentoo, наравне с вечными фаворитами Debian,

Slackware, Fedora, Mint, Ubuntu и т.д. Gentoo и Arch учат, как устроен Linux, лучше любого другого дистрибутива. Но неопытные пользователи Linux сторонятся их, и не без причины. Эти двое не для тех, кто никогда раньше не пробовал Linux.

В зависимости от вашего прошлого опыта работы с компьютером, вы можете посчитать пять

дистрибутивов из нашего списка слишком простыми. Тогда выбирайте Fedora, Slackware, Debian, или любой другой.

Если вы уже совсем зеленый новичок, лучше будет начать с одного из предлагаемой нами великолепной пятерки, а уже потом переключаться на какой-то другой. **LXF**

Влейте свой голос в Глас народа

Зачем вам спонсор, ищущий от поддержки ваших идей выгоду? Маянк Шарма советует: пусть лучше скинутся энтузиасты вашего проекта.

Какая связь между умными часами с индивидуальной настройкой, игровой консолью на Android, платформой для блоггеров с открытым кодом и видеоигрой на выживание в открытом мире? Помимо их огромной популярности в своих жанрах, все эти захватывающие проекты по оборудованию и ПО сумели появиться на свет исключительно благодаря небольшому финансовым вливаниям от совершенно посторонних людей.

Идея общественного финансирования [crowdfunding] или обращения за пожертвованиями к отдельным людям не нова. Новым стал способ сбора денег. Благодаря Интернету и социальным сетям стало намного проще находить друзей по интересам и связываться с ними. И это означает наличие более эффективных способов налаживания контактов с людьми, готовыми открыть кошелек для оказания финансовой поддержки вашим проектам.

Общественное финансирование — настоящая экономическая революция. По данным исследовательской фирмы Massolution, в 2012 году сумма общественного финансирования достигла \$2,7 миллиардов, и профинансировано

было более миллиона компаний. К концу 2013-го, по расчетам фирмы, «инвесторы с улицы» доведут эту цифру аж до \$5,1 миллиарда.

Но хотя общественное финансирование позволяет вам просто купаться в деньгах, не стоит нырять в него бездумно. Не обольщайтесь историями супер-успешных кампаний. Использование потенциала толпы — сложный процесс. Сбор средств не исчерпывается просто публикацией своей идеи с предложением бесплатных чудес — вряд ли шальные миллионы так и хлынут на ваш банковский счет в ответ.

Прежде чем приступить к общественному сбору средств, придется немало узнать. Успешный проект общественного финансирования требует наличия всех атрибутов социальной и деловой кампании. Вам придется дополнить свою потрясающую идею исследовательскими и маркетинговыми стратегиями.

В нашей статье мы расскажем обо всех подробностях общественного финансирования и подчеркнем его преимущества перед традиционными способами получения финансовой поддержки.

Также мы предложим вам советы от тех, кто успешно этим занимался и готов помочь вам спланировать ваш проект полностью, от запуска до успешного финансирования и реализации, избегав подводных камней. Вы также узнаете о разных платформах общественного финансирования, чтобы принять вполне взвешенное решение при выборе наиболее пригодной для финансирования вашей личной кампании.



Молодым бойцам

Экспресс-курс сбора нала.

Участие в чьей-то кампании общественного финансирования не требует почти ничего сверх щелчка мышью, но запуск такой кампании — работа серьезная. Если вы подумываете запустить свою кампанию, не особо потратив на нее время, вспомните такие цифры: успеха на Kickstarter добились только менее 50% всех проектов, а 10% не получили вообще ни гроша!

Но, ухватив суть общественного финансирования, вы сможете перевернуть эту статистику. Вы поймете, что это лучшая методика получения капитала и завоевания места на рынке, пытаетесь ли вы запустить в массовое производство инновационное устройство или вместе создать некую программу.

Что такое краудфандинг

Это общественное финансирование, подмножество более широкой концепции краудсорсинга [crowdsourcing], что, в свою очередь, может много чего означать, но вовлекает процесс доступа к большой группе людей через Интернет, чтобы использовать их знания, умения, время или другие ресурсы.

Краудфандинг — частный случай, касающийся сбора личности; это коллективное сотрудничество людей, собирающих средства на формирование финансовой поддержки некоего проекта. Человек или группа людей предлагают план проекта, требующего средств, а затем собирают группу людей, готовых выложить деньги на разработку и реализацию. При этом не обойтись без возможности массового доступа, обеспечиваемой Интернетом: люди, имеющие определенные таланты и идеи, выходят на тех, кто заинтересован в идее и обладает фондами для инвестиций.

У проектов общественного финансирования есть ряд характеристик, позволяющих отличить их от других способов получения спонсорства и сотрудничества по проекту. Для начинающих проекты общественного финансирования включают микрофинансирование незначительными суммами. Таким образом снижается порог участия в проектах. Те, кто раньше и не воображал себя инвесторами, теперь вполне могут стать ими.

Проекты общественного финансирования также обязаны предлагать некое вознаграждение, хотя бы и нематериальное: например, статус участника проекта или шанс ощутить себя «хорошим». Вознаграждения привлекают к проекту людей, а разнообразие рода вознаграждений и стимулов, часто в зависимости от вложенной суммы, делает проект привлекательнее для еще более обширных слоев населения.

Также все проекты должны иметь четко сформулированные цели — в основном это сумма, которую желательно собрать за определенное время. Другой ключевой аспект проектов общественного финансирования — тщательно проработанные реклама и маркетинг.

Традиционно финансирование ожидалось от венчурных компаний и «добрых ангелов» — частных инвесторов, или просто в виде старого доброго займа из вашего местного банка. Энергичный предприниматель обращается к этим источникам с бизнес-планом, и ему остается надеяться, что скрупулезно изучающий

Популярные платформы краудфандинга

Итак, с беготней покончено, и вы готовы обратиться к людям за финансовой поддержкой. Где это сделать? Надо ли вам создавать свой сайт со сложной инфраструктурой по обработке кредитных карт или PayPal? Почему бы и нет (см. список *Самопальные платформы краудфандинга*, который начинается на стр. 39); но намного проще начать свою кампанию на уже готовой платформе.

Готовая платформа дает вам ряд преимуществ. Во-первых, в ней уже есть функции, необходимые для проведения кампаний. Она разместит все ваши мультимедиа и предложит инструменты, необходимые для вовлечения нужной вам аудитории. Кроме того, вы получите доступ к инструментам разных социальных сетей, что поможет продвижению вашего проекта. Платформа также донесет ваш проект до тех, кто ищет, куда бы всадить свои денежки.

Помните также, что вам выставят счет за использование их сервисов, а иногда и предложат оплатить сбор за механизм, который вы применяете для сбора нужных вам средств. Вот выбранные нами платформы.

Kickstarter

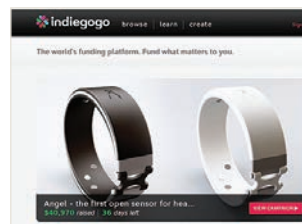
Сайт: www.kickstarter.com
Модель финансирования: Все или ничего
Сборы: 5%, если цель достигнута, + сбор за обработку платежей
Способ вклада: Amazon Payments



Справка: 800-фунтовый монстр общественного финансирования. Размещает только кампании определенных категорий и предусматривает процесс одобрения. Также здесь есть куча информации для создания и успешного проведения кампании.

Indiegogo

Сайт: www.indiegogo.com
Модель финансирования: Все или ничего
Сборы: 4%, если цель достигнута, 9% — если нет, + сбор за обработку информации
Способ вклада: PayPal, ACH, Wire Transfer или FirstGiving



Справка: Размещает кампании со всего мира, по всем категориям. Также предлагает большое количество документации. Если вы не достигнете своей цели, вам встанет дороже.

Fundable

Сайт: www.fundable.com
Модель финансирования: Все или ничего
Сборы: \$99 в месяц во время активной кампании + сбор за обработку информации
Способ вклада: WePay



Справка: Платформа, специально разработанная для финансирования стартапов, способных предложить некое вознаграждение и стимул за финансирование. Как сказано на сайте, начинающие проекты, предлагающие долю участия, обычно собирают до \$10 миллионов.

каждую деталь их плана человек уверует в быстрый возврат любой начальной инвестиции.

Это составляет резкий контраст общественному финансированию, когда с помощью социальных сетей предприниматель может получить через сайты доступ к сотням потенциальных инвесторов, которые вовсе не обязательно заинтересованы в доении чьих-то талантов. Именно это и делает его одним из самых популярных методов получения финансовой поддержки.

Барьеров больше нет

В отличие от обычных методов финансирования, когда деньги поступают от одного предприятия или тесно сплоченной группы, кампания по получению общественного финансирования для продвижения продукта на массовом рынке и покрытия финансовых затрат на продукт использует социальные сети. Таким образом небольшие пожертвования от множества людей позволяют в конечном итоге получить солидную сумму денег.

Самое потрясающее в краудфандинге — это видеть его в действии: «Главное в краудфандинге то, что он полностью устраняет барьеры между человеком, у которого есть некая идея, и его клиентом», полагает Джеймс Кэри [James Carey], дизайнер в Big Robot, разработчик игры *Sir, You Are Being Hunted*, профинансированной с помощью кампании на Kickstarter (www.kickstarter.com).

«Убеждать в том, что игра стоит свеч, приходится только конечного пользователя. Никаких банков и фокус-групп, никаких советов по выдаче грантов. И тогда идеи, которые ни один „эксперт“ в определенной индустрии не считал бы „стоящими“, получают шансы на жизнь. По-моему, это лучшее, что за все эти годы сделал Интернет», говорит Кэри.

В зависимости от выбранного вами для работы типа кампании, люди, которые и создают общественное финансирование, не будут ожидать от вас финансовой компенсации, доли в деле или даже возврата средств. Не рассматривая это как традиционную форму инвестиций, они делают вложения, чтобы поддержать проект, в который верят, в надежде, что он добьется успеха. Кэри считает, что общественное финансирование стоит считать скорее приобретением, чем инвестицией: «Люди делают гипотетические покупки. Покупают авансом. Делают предзаказ».

Еще один фактор, отличающий краудфандинг от традиционных источников финансирования — чувство соучастия. Вы получаете средства от тех, кто увлечен вашим проектом или идеей и хочет помочь реализовать их.

На своем вебинаре одна из старейших платформ общественного финансирования, Indiegogo (www.indiegogo.com), рекомендует считать общественное финансирование проявлением коллективного энтузиазма, а не попрошайничеством. Идея успешной

кампании общественного финансирования должна заключаться в том, чтобы другие разделили вашу увлеченность проектом, чтобы аудитория заинтересовалась им.

Вы позволяете другим заглянуть за кулисы и понять, почему вы так увлечены проектом, чтобы они тоже вступили в команду и оказали вам содействие. Неважно, много у них денег или нет. Если они не разделят вашего энтузиазма, они не дадут ничего.

Побочные фишки

Хотя самой очевидной причиной краудфандинга является получение средств, есть и несколько иных причин, по которым к этому способу стоит прибегнуть ради своего проекта. Успешная кампания краудфандинга может дать вам фантастическую возможность привлечь к себе внимание и заинтересовать своим проектом. Вы не только сможете оценить спрос на свой проект, но и сделаете это с минимальным финансовым риском за очень короткий промежуток времени по сравнению с возможностью привлечения венчурной компании за счет доли в своем проекте. Кампания Ubuntu Edge на Indiegogo — классический тому пример. Canonipal не достигла своей увесистой цели в \$32 млн (но и получив \$12,8 млн, кампания побила все рекорды краудфандинга), что не помешало популярной прессе разглагольствовать о появлении Ubuntu OS на рынке смартфонов в 2014 году.

В целом, кампания краудфандинга позволяет оперативно оценить, насколько хороша ваша идея и есть ли спрос на нее на рынке. Неудивительно, что многие менеджеры успешных компаний вернулись к венчурным компаниям, изначально им отказавшим, чтобы продемонстрировать наличие спроса на свой проект и готовности людей заплатить установленную ими цену. И разговор шел уже на совершенно ином уровне.

Модели краудфандинга

Все, что вы уже прочитали — лишь эскизное описание краудфандинга. На практике краудфандинг объединяет несколько разных моделей финансирования.

Для начала, есть модель пожертвований, существующая веками. Люди жертвуют деньги на проект или идею, так как верят, что у него есть моральная и этическая ценность и это хорошо для общества. Они не ожидают никакого возврата своих финансовых пожертвований, кроме морального удовлетворения.

Есть также модель, основанная на вознаграждении. Именно она при мысли о краудфандинге большинству приходит на ум прежде всего. В этой модели спонсоры могут авансом приобрести продукт или получить некое вознаграждение. То есть общество делает финансовое вложение в проект, а проект предлагает ему нечто взамен.

Эта модель далее подразделяется на две модели.

Законы о краудфандинге в Великобритании и США

Чтобы уяснить юридические обоснования краудфандинга, принятые в Великобритании, нам пришлось обратиться к нашему домашнему юрисконсульту Шашанку Шарме.

Первая в Великобритании регламентированная платформа краудфандинга, Abundance Generation, была одобрена в 2011 году, а запущена в 2012. Тогда платформы краудфандинга в Великобритании регламентировались квазисудебным органом, известным как Financial Services Authority (FSA). С апреля 2013 эту задачу передали в руки вновь

учрежденному Financial Conduct Authority (FCA). Согласно FCA, у инвесторов, принявших участие в краудфандинге, нет реальной защиты от мошенников. Хотя в случае мошенничества инвесторы могут обратиться к Financial Ombudsman Service (FOS) или Financial Services Compensation Scheme (FSCS), эти сервисы доступны только в случае, если платформа была авторизована FCA или его предшественником, FSA.

Однако есть несколько платформ, работающих без подобной авторизации, и по сведениям FCA,

количество неавторизованных платформ намного превосходит количество утвержденных.

В США краудфандинг не имел юридической силы до принятия закона JOBS (Jumpstart Our Business Startups) в 2012 году. Не так давно, в сентябре 2013 года, в силу вошел Раздел II закона JOBS. Закон JOBS позволяет начинающим компаниям получать финансирование на общественных началах, а Раздел II дает им право использовать для оповещения о своей кампании такие сайты, как Twitter и Facebook.

Первая — это модель «все или ничего», когда проект ставит перед собой конкретную финансовую цель, которую и стремится достичь. Если кампания не достигает цели, инициаторы проекта не получают финансирования, а заказчики не получают своего вознаграждения.

Другая модель — «забирай все», когда все средства, собранные во время кампании, остаются у инициаторов проекта, независимо от того, достигли они своей цели или нет. Многие считают модель «все или ничего» более привлекательной.

Есть еще менее известная модель «от щедрот [bounty]», когда все собранные на определенную задачу средства передаются тому, кто берется закончить проект.

И еще одна менее возвышенная, но революционная модель — модель «акций», которая позволяет людям вносить небольшие суммы на финансирование начинающих компаний-стартапов [startup]. Взамен они рассчитывают на дивиденды за счет прибыльности бизнеса. Она более популярна в Великобритании и Европе, поскольку там меньше юридических препятствий (см. *Законы о краудфандинге в Великобритании и США*), но скоро эта модель может быть легализована и в США.

Как сделать кампанию успешной

Есть немало отличий, которые и отделяют успешную кампанию от провальных. Перед началом кампании как можно подробнее продумайте все ее аспекты. Сидя на попе ровно, взлететь не получится. Вам следует обеспечить план на каждую неделю вашей кампании.

Джеймс Кэри говорит, что ваша подготовка является также и показателем, что вы уже вложили в проект часть своего времени и денег: «Мы не хотели выходить на Kickstarter, пока у нас не появился солидный прототип, основанный на полугоде работы и солидном вложении денег, которое эта работа подразумевала. Если вы сами не готовы рискнуть своими вложениями ради проекта, то почему это должны делать другие?»

Одним из первых вы должны продумать заявку вашей кампании. Просмотрите успешные кампании и выясните, почему они успешны. Большинство заявок очень индивидуальны и ясно показывают, кто стоит за проектом и какие проблемы они надеются решить с его помощью. Кроме того, они показывают потенциальным спонсорам, как они могут принять участие в проекте.

«Заявки успешны, если люди решают, что они добьются успеха, — говорит Кэри. — То, что выглядит способным заработать много денег, обычно их и получает. Это игра восприятия».

Хорошая заявка представляет кампанию как в письменном виде, так и в форме видео. В соответствии со статистикой блога Indiegogo, кампании с видео собирают в два раза большие суммы, чем кампании без видео. Также помните, что видео успешной кампании длится минуты 3.

Обязательно также ставить выполнимую цель. Не озвучивайте некую произвольную цифру. Сначала надо заняться серьезными расчетами и установить консервативную сумму, способную продвигать ваш проект и выполнить обещания по вознаграждениям.

На Indiegogo 87% кампаний, которые достигли своих целей, превзошли их в среднем на 32%. Советую проектам ставить разумные цели, Кэри отмечает, что, похоже, существует некий шаблон: «На самом деле, наблюдается любопытное правило: успешные проекты Kickstarter в основном достигают 150% поставленных целей. Это предполагает, что если вы хотите собрать £60К, стоит просить £40К, и вы получите свое с большей вероятностью».

Вы также должны очень умно изобрести вознаграждения. Сделайте их уникальными, чтобы люди им радовались и рассказывали о них друзьям, но не взламывайте банк. И опять же,

Популярные платформы краудфандинга

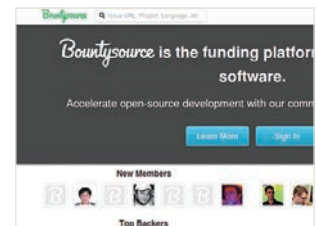
(продолжение)

Bounty Source

Сайт: www.bountysource.com

Модель финансирования: От щедрот или Все или ничего
Сборы: 10% невозмещаемого взноса за размещение премий
Способ вклада: Paypal, чек, Google Wallet, телеграфный перевод

Справка: Предназначена для помощи программам с открытым кодом. Копит индивидуальные выплаты на решение общих проблем и просьбы о создании функций, использовалась также для сбора денег на крупные обновления или новые проекты.



Freedom Sponsors

Сайт: <http://freedomponsors.org>

Модель финансирования: От щедрот
Сборы: 3% + сбор за обработку информации
Способ вклада: Paypal и Bitcoin

Справка: Позволяет многим пользователям внести свою лепту. Премии выплачиваются только когда спонсоры удостоверят выполнение работы. Использует параллельные платежи Paypal для распределения премий между разработчиками.



Catincan

Сайт: www.catincan.com

Модель финансирования: Все или ничего
Сборы: 10% после получения установленной суммы
Способ вклада: Paypal, Bitcoin, безналичный перевод

Справка: Кампании разрешается запускать только разработчикам существующих проектов с открытым кодом. Отображаются все подробности, и у разработчиков есть 60 дней на сбор целевой суммы, в противном случае деньги остаются у спонсоров.



BountyOSS

Сайт: <https://bountyoss.com>

Модель финансирования: От щедрот
Сборы: 5% + сбор за обработку информации
Способ вклада: Кредитная карта или банковский счет

Справка: Разработана для сбора средств у компаний и других коммерческих организаций, использующих программы с открытым кодом. Кампании разрешается запускать только разработчикам, участвующим в проектах с открытым кодом.



неплохо познакомиться с опытом предыдущих успешных кампаний, у которых было от пяти до восьми наград.

Лучше всего предлагать широкий ассортимент наград. Вместо выкатывания их всех одновременно, лучше будет обновлять их по ходу кампании. И опять же, наилучшим способом будет посмотреть на другие кампании в поисках самых популярных фишек. Джон Нолан [John Nolan], который успешно профинансировал свою платформу блоггинга с открытым кодом Ghost, поделился своим рецептом успеха: «Притягательная идея, некая ценность в качестве вознаграждения за вклад и ясная формулировка того, что может предложить создатель проекта».

Создайте сообщество

Выработав все эти важнейшие элементы, потратьте некоторое время на создание плана коммуникаций. Создатели успешных кампаний вели активное общение в течение всей своей кампании и постоянно подогревали интерес инвесторов. Хотя многие считают платформы краудфандинга чем-то вроде блога, большинство их размещает одно-два обновления в неделю. Там будут обсуждаться обновления, пройденные этапы и новые вознаграждения по мере их появления.

Джеймс Кэри полагает, что обновления — ключ к успеху кампании, и добавляет: «Вы должны наладить хорошие отношения со своими спонсорами, давая им знать, что творится за кулисами. Краудфандинг — это патронаж, но патронажу сопутствует некое ощущение сопричастности. Вы здесь вместе со своими спонсорами; пусть же они об этом знают».

И, наконец, главное: не забывайте оттачивать навыки работы в социальных сетях. Вам нужно добиться огромного количества мелких пожертвований, которые вместе составят солидную сумму. Для этого нужно суметь достичь большой аудитории и познакомиться ее с проектом, а социальные сети являют собой отличный инструмент для расширения границ вашей кампании.

Используя социальные сети и идею шести степеней разделения, вы можете рассчитывать, что члены изначального ядра вашего сообщества донесут информацию уже до своих сообществ и в конечном итоге создадут заразительную кампанию по распространению.

Помните также, что крайне важны первые дни кампании. И исключительно вы сами отвечаете за придание ей начального импульса. Вот почему так важно определить свою первичную целевую аудиторию — ту, что запустит вашу кампанию и станет ее спонсорами и промоутерами.

Краудфандингу сопутствуют некоторые потенциальные ловушки. Поэтому перед запуском своей кампании следует поучиться на ошибках провальных кампаний, чтобы не отпугнуть инвесторов и клиентов.

По словам Тони Франса [Tony França], создателя FreedomSponsors, есть два фактора успеха кампании краудфандинга: «Некоторые проекты приходят к провалу потому, что у них нет рынка,

а некоторые — потому, что их маркетинговая политика недостаточно хороша. Ваша идея должна предлагать то, что действительно нужно людям или что они хотят. В противном случае рынка у вас нет, и ваш проект обречен. Точка. И даже если у вас есть рынок, нужно работать над маркетингом. Мир должен знать о вашей кампании. Это самая трудная часть. Я считаю, что самые успешные кампании — это те, которые так успешно продвигают свой продукт на рынке, что людям хочется рассказать о нем своим знакомым, и тогда они заражают».

Уоррен Конкель [Warren Konkel], руководитель Bountysource, указывает еще на одну причину неудач тех кампаний, где не были проделаны предварительная работа и исследования: «В общем случае те, кто искал финансирования, не достигли цели по одной из двух причин: или людям это было не нужно, или люди этого просто не поняли».

Однако еще бóльшая проблема в том, что проекты плохо оценивают раскладку своих финансов. Не проведя изначального исследования, не удивляйтесь, если общая стоимость проекта и вознаграждений окажется больше собранной суммы.

Одним из ключевых факторов, которые приводят к неудаче кампаний, по словам Нолана, являются малоценные награды с чрезмерными критериями выполнимости: «Вы в самом деле намерены лично подписать 10 000 открыток? Да ну?» Запуск кампании и постановка цели, когда вы полагаете, что сможете «продумать всю эту ерунду с налогами» позднее — еще одна проблема, которую он выделяет.

Кэри соглашается, что некоторые проекты обещают слишком много, не просчитывая должным образом свои возможности по цене вознаграждений: «Я полагаю, стоимость доставки привела к банкротству не один проект, потому что они не учли, сколько фирменных кружек им придется отправить».

Грустный пример — история с Hanfree iPad Accessory. Создатель проекта недооценил реальную стоимость вывода своей идеи на рынок и массового производства продукта. Собрав целевую сумму \$35 000, он не сумел предоставить ни продукт, ни вознаграждение. И в итоге один из раздосадованных спонсоров предъявил создателю проекта иск, доведя его до банкротства.

Готовы двигаться?

Сейчас вам уже, должно быть, не терпится приступить к своему драгоценному проекту. Но сначала вот вам краткий обзор действий, обязательных при запуске и в работе хорошей кампании краудфандинга. Есть ряд моментов, которые надо решить даже еще до того, как вы элементарно начнете искать платформу для своей кампании.

Сперва подсчитайте, какой цели вы хотите достичь, не забывая при этом о природе, количестве и разнообразии вознаграждений. Также потратьте время на определение срока вашей кампании, оглядев подобные ей успешные проекты. Простым ответом может показаться срок побольше, но помните, что вам придется

Идеальный срок для кампании

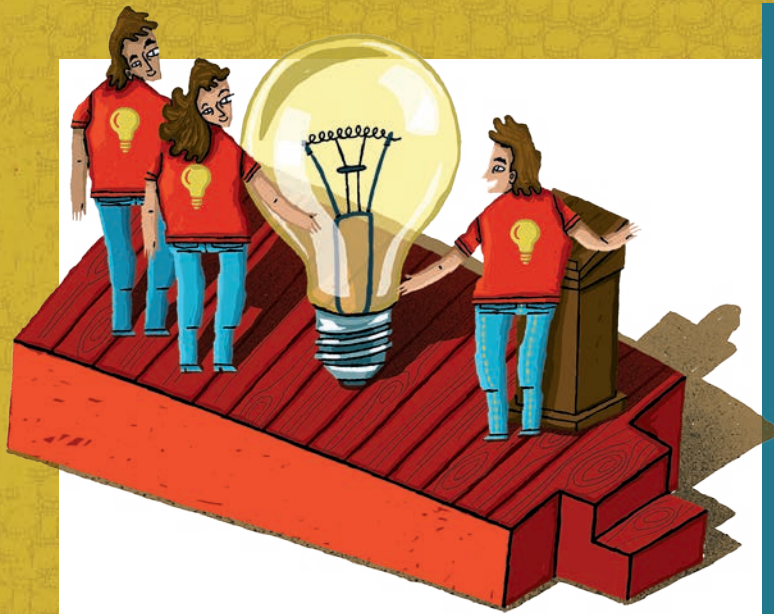
Связь между длительностью кампании и суммой денег, которую она собирает, существует — но она не такова, как вы думаете. Если вы думаете, что чем дольше работает кампания, тем больше денег она вам раздобудет, то вы ошибаетесь.

По данным Indiegogo, большинство успешных кампаний работает в среднем от 30 до 40 дней. Тогда кампания достаточно коротка, чтобы было

видно ее окончание, и в то же время достаточно долгая, чтобы раскрутиться и задействовать широкую аудиторию.

Исследования Kickstarter указывают на тот же самый вывод: проекты на Kickstarter могут существовать только 60 дней, потому что, согласно его статистике, более длительные проекты редко бывают успешны.

Рекомендуется дней 30 или меньше, потому что меньшая длительность создает полезное ощущение срочности проекта. Кроме того, при более короткой кампании вы сможете более эффективно подготовиться и провести ее: работа с кампанией отнимает много времени, и для большинства проектов это практически полная занятость в течение всего срока кампании.



«Вы должны наладить хорошие отношения со спонсорами.»

следить за кампанией и придавать ей импульс на всей ее протяженности (см. *Идеальный срок для кампании*, стр. 38).

Найдя ответы на эти вопросы, начинайте искать платформу. На последних страницах нашей статьи мы рассмотрим ряд лучших платформ и инструментов общественного финансирования. Решите, какая модель финансирования наилучшим образом подходит вам и вашему проекту.

Планирование продвижения

Приступив к работе, подумайте также о разных инструментах социальных сетей, применимых для продвижения вашей кампании. Широкое использование разных социальных сетей может показаться хорошей идеей, однако это стоит делать, только если вы в состоянии их отслеживать. Скажем, оставшийся без ответа вопрос от потенциального инвестора может причинить кампании немалый вред.

Составьте также список своей первичной целевой группы. Помимо контактов в своей сети, помните о необходимости выходить на популярные сайты и блоги, аудитория которых имеет те же интересы. Было бы также неплохо проинформировать их о вашей кампании за пару дней до ее запуска — и еще раз, когда она уже будет работать.

Не забывайте исследовать разные возможности продвижения вашей кампании даже после того, как вы ее уже запустили. Платформа краудфандинга — это не eBay. Ваша кампания не сможет сама себя продать. Помните также, что вы ищете не только тех, кто поддержит ваш проект финансово; вы также ищете тех, кто поможет вам его продвигать в дальнейшем в своих сетях.

Постоянно держите своих инвесторов в курсе всех последних новостей. Делитесь с ними своими впечатлениями, новыми идеями, новыми наградами, и можете даже спрашивать у них совета или интересоваться их мнением там, где это нужно. Что бы вы ни делали, пусть они участвуют в этом. Получив финансирование, сообщайте своим инвесторам о состоянии их вознаграждений, да пошевеляйтесь, чтобы реализовать свой проект.

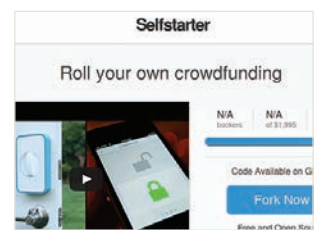
Самопальные платформы краудфандинга

У проектов бывает целый ряд причин избегать существующих платформ краудфандинга, предпочитая создавать собственные. Для некоторых причина кроется в идеологии. Вы можете себе представить, чтобы FSF проводил кампанию общественного финансирования на проприетарной платформе с закрытым кодом?

Но есть и чисто прагматические причины. Чем больше проектов на платформе, тем меньше шансов их толком разглядеть. Поскольку краудфандинг обретает признание как легитимный механизм сбора средств, при собственной платформе вы будете собирать средства помимо уже имеющихся платформ. Один из самых удачных примеров собственной платформы для кампаний — Lockitron. Этот проект был отвергнут Kickstarter, но они создали собственную инфраструктуру краудфандинга — и собрали более \$2 миллионов. Затем они выпустили свою платформу с открытым кодом, чтобы ею могли пользоваться и другие.

SelfStarter

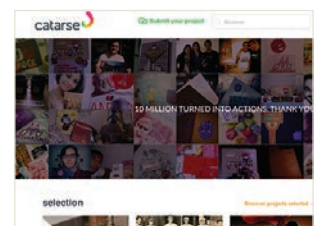
Сайт: <http://selfstarter.us/>
Лицензия: MIT License
Сборы: бесплатная
Способы вклада: Amazon Payments, Stripe, WePay



Справка: Разработанная на Ruby on Rails, эта платформа расширяема с помощью кода специальной аутентификации, администрирования и управления продуктом. Разработчики предлагают использовать ее на платформе приложений Heroku.

Catarse

Сайт: <https://github.com/catarse>
Лицензия: MIT License
Сборы: Бесплатно
Способы вклада: MoIP, Paypal



Справка: Еще одна программа на Ruby on Rails. Можете использовать ее при создании собственной платформы краудфандинга для размещения нескольких кампаний. На ней работает бразильская платформа <http://catarse.me/en>. Имеет минимум документации, но активный список рассылки.

Goteo

Сайт: <https://github.com/Goteo>
Лицензия: GNU AGPL v3
Сборы: Бесплатно
Способы вклада: Paypal



Справка: Программа написана на PHP и облегчает общение между пользователями и менеджерами кампании, а также создание блогов и статических страниц. Кроме англоязычного руководства по установке, вся остальная документация разработчика — на испанском.

Краудфандинг программ с открытым кодом

Благодаря природе свободного ПО и ПО с открытым кодом, его разработки в основном ведутся через краудсорсинг, а светит ли им также и краудфандинг?

Как выясняется, самые популярные модели краудсорсинга представляют собой уникальные задачи для программ с открытым кодом. Хотя вы можете найти успешные кампании для ПО с открытым кодом на популярных платформах общественного финансирования, таких, как Kickstarters, все же они там не совсем естественны.

В интервью opensource.com Уоррен Конкель, руководитель Bountysource, платформы краудсорсинга, разработанной специально для финансирования ПО с открытым кодом, говорит, что свободному ПО и ПО с открытым кодом нужен совсем другой подход, нежели обычным предложениям на других популярных платформах.

«Прочие платформы хорошо работают с физическими потребительскими товарами и технологиями, но мы полагаем, что программам с открытым кодом нужна лучшая модель финансирования, которая больше соответствует способу создания программ».

Беседа с LXF, Конкель развивает свою мысль, говоря, что одна из причин, которая отделяет разработку ПО от разработки оборудования, в том, что разработка ПО печально знаменита крайней сложностью оценки ее стоимости. «Платформа Bountysource решает эту проблему, напрямую соединяя желающих получить финансирование с запросами на устранение ошибок и разработку функций».

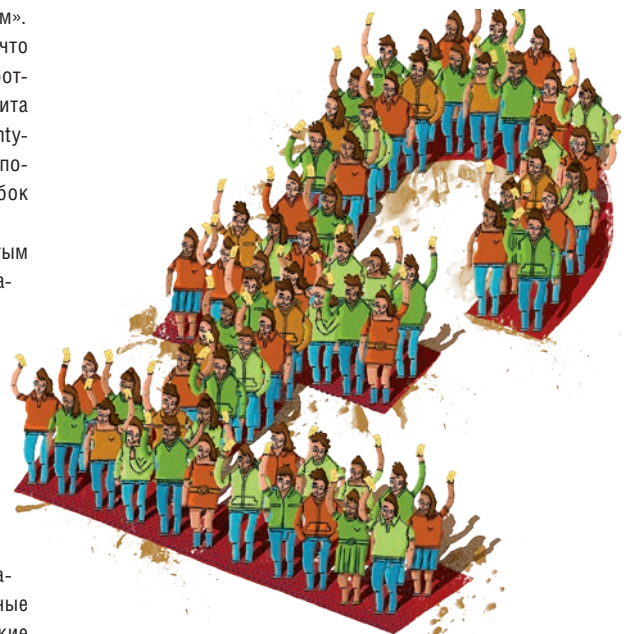
Еще одна проблема для краудфандинга программ с открытым кодом, решивших использовать платформы с моделью вознаграждения — выбор правильного вознаграждения, способного привлечь спонсоров. Типичная кампания проприетарного ПО может предложить в качестве вознаграждения разные версии программы. В противоположность им, ПО с открытым кодом по природе своей не может предложить версии продукта с функциями исключительно для своих спонсоров.

Они могут предложить дополнительные сервисы — например, персональную поддержку. Некоторые отмечают своих спонсоров на своих сайтах, некоторые предлагают эксклюзивный контент, например, постоянные новостные бюллетени исключительно для спонсоров, и даже физические

продукты: например, эксклюзивные футболки с красочным отображением тематики кампании.

Как и всегда, знакомство с предложениями других кампаний открытого ПО может привести вас на мысль о том, что можете предложить вы. Платформа блоггинга Ghost предлагала бесплатные учетные записи на своем сервисе хостинга, обещая спонсорам доступ к платформе на начальной стадии и предлагая им возможность зарезервировать свое имя пользователя на сайте сообщества, которое отображалось со специальной эмблемой, чтобы выразить признательность за их поддержку.

Многие также считают краудфандинг в разработке ПО с открытым кодом частью долгосрочного процесса, а не разовой инвестицией. Конкель говорит, что на Bountysource стремятся создать долгосрочные взаимоотношения между разработчиками



Люби и знай профессиональный жаргон

Желая запустить кампанию краудфандинга, вы постоянно будете наткаться на эти термины.

Кампания [Campaign]: Проекты, под которые вы хотите получить финансирование.

Спонсоры [Backers]: Могут также именоваться «Вкладчики [Contributors]» и «Инвесторы [Funders]»; это частные лица, которые поддерживают вашу кампанию, предоставляя вам деньги.

Цель [Goal]: Денежная сумма, которую кампании требуется собрать за предварительно установленный срок.

Расширенные цели [Stretch goals]: Если кампания достигает своей цели раньше срока, в ней может появиться дополнительный пункт, над которым будет вестись работа, когда будет достигнута новая, более высокая цель.

Залог [Pledge]: В модели «все или ничего», пока проект не достигнет своей цели, все денежные взносы именуется залогом.

Вознаграждения [Rewards]: Именуемые также «Подарками [Gifts]» или «Бонусами [Perks]», это стимулы, предлагаемые спонсорам в обмен на предоставление ими финансовой помощи продвигаемому кампанией проекту.

и инвесторами: «Когда некто оказывает поддержку тому, кому она нужна, есть шансы на то, что они и впоследствии поддержат другой проект или создадут некую премию».

Филип Хоргер [Philip Horgger], активный организатор краудфандинга на FreedomSponsors, еще одной платформе краудфандинга для ПО с открытым кодом, делится своим интересным опытом подобных взаимоотношений разработчик—жертвователю.

Хоргер запустил финансирование для улучшения пользовательского интерфейса LibreOffice. Член команды разработки LibreOffice как-то пришел к нему и сказал, что спонсорство такой широкомасштабной проблемы, конечно, весьма многообещающий знак для разработчиков, но это не самый удобный способ производить улучшения UI. Вместо этого разработчик порекомендовал Хоргеру спонсировать более частные улучшения, например, устройство выбора цвета и т. п. «И я разделил свое предложение на более конкретные задачи. В конечном итоге финансирование было тем же, но его распределили более разумно».

Лучше, чем пожертвование

Денежные пожертвования от пользователей для поддержки разработки программ с открытым кодом — вещь не новая. Однако краудфандинг приносит куда большую радость, чем просто пожертвование.

Хоргер говорит, что финансирование на FreedomSponsors изначально более тонкое и личностное, чем пожертвования, поскольку предложение связано с определенным и достижимым результатом: «Вы ощущаете свое участие и свою ответственность за успех финансирования. Намного приятнее и солиднее, когда вы можете указать своему соседу или приятелю на некую функцию в какой-то популярной программе и сказать: «Я могу ее реализовать»».

Другое преимущество таких платформ, как FreedomSponsors, по словам Хоргера, совершенно лишаящее смысла традиционные пожертвования, заключается в возможности финансировать собственные проекты. Сначала это может показаться странным, но, как любит говорить Хоргер, «в моем безумии есть свой метод».

«Финансирование — это способ для меня мотивировать/вознаградить внешнюю разработку, и оно включает необходимость начинать с малого. Таким образом я могу собрать активное и здоровое сообщество инвесторов, которое не только снимет с моих плеч бремя программирования и поддержки в перспективе, но также означает, что мои проекты будут жить дальше, если я вдруг попаду под автобус».

Будущее краудфандинга

Подобно своим товарищам, Тони Франса из FreedomSponsors весьма оптимистичен насчет будущего краудфандинга и того, как оно может помочь разработчикам создавать фонды: «Мы являемся свидетелями рождения нового бизнеса, финансируемого на общественных началах. Я верю, что краудфандинг как способ получения средств станет набирать все большую силу».

Конкель также признает модели сбора средств, особенно для тех проектов, которые требуют наличия авансового капитала для достижения экономии за счет масштаба, и где вознаграждения по сути являются предпродажами. Однако он полагает, что это будет работать не для всех. «Мы считаем, что будущее краудфандинга в открытом коде — это премиальная модель: размещение индивидуальных премий на существующих заявках на исправление ошибок и разработку конкретных функций. Очень быстро появятся те, кто будет отслеживать эти премии, их число вырастет, и премии позволят спонсорам концентрировать внимание разработчиков на нужных им проблемах». **LXF**

Самопальные платформы краудфандинга

(продолжение)

GitTip

Сайт: www.gittip.com
Лицензия: CC license
Сборы: Бесплатно
Способы вклада: Bitcoin, кредитные карты



Справка: GitTip — платформа краудфандинга для еженедельных пожертвований. Вы можете использовать ПО для настройки собственного еженедельного обмена подарками для участников вашего проекта или сообщества.

Spot.U.s

Сайт: <https://github.com/spot-us>
Лицензия: GNU GPL
Сборы: Бесплатно
Способы вклада: Paypal



Справка: Этот проект Ruby on Rails позволяет настроить сайт краудфандинга, помогающий частным лицам или группе собрать средства на вознаграждение журналистов-фрилансеров. Проект имеет минимум документации, но его можно увидеть в действии на www.spot.us.

Crowd-Hoster

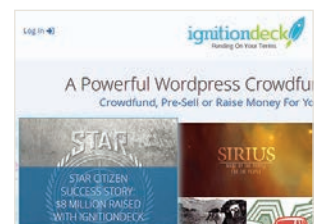
Сайт: www.crowdhoster.com
Лицензия: MIT License
Сборы: Бесплатно
Способы вклада: Кредитные карты, Bitcoin



Справка: Гордо именует себя «WordPress для краудфандинга», проекты пока размещаются только по приглашению. Она бесплатна и весьма бесхитростна в настройке и администрировании, поскольку ничего устанавливать не надо. Включает инструменты взаимодействия с инвесторами.

Ignition Deck

Сайт: <http://selfstarter.us/>
Лицензия: Проприетарная
Сборы: От \$79
Способы вклада: Paypal, Stripe, WePay

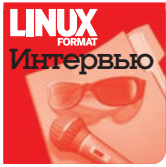


Справка: Фактически, этот инструмент — плагин WordPress. Доступен в нескольких версиях, начиная от \$79. Вы можете установить этот плагин на разные версии WordPress и приобрести дополнительные темы, например, платежные шляпы и панель аналитики.

Роберт 'r0ml' Лефковиц

Человек Возрождения

Грэм Моррисон встретился с Робертом 'r0ml' Лефковицем — человеком, способным уместить 2000 лет философии Open Source в 3000 слов.



Роберт Лефковиц [Robert Lefkowitz], почти всегда именуемый просто 'r0ml' — компьютерщик, разработчик и мыслитель старой закалки, переключившийся с ядер-

ной физики на программирование в 1970-х. Однако чем больше слов мы здесь напечатаем, тем меньше останется ему, так что поехали.

Linux Format: Считаете ли вы, что компьютеры и языки программирования должны стремиться к большей понятности?

Роберт Лефковиц: Так рассуждал еще Карл Великий, издавая в 789 году капитулярный Admonitio Generalis [Всеобщее увещание], где утверждалось, что монахам следует учиться читать, чтобы понимать слово Божие и вернее следовать его заветам. Если цель — в этом, почему бы не обучить этому каждого мальчика? И действительно, в капитулярии говорится о создании школ для всех детей мужского пола, как свободных граждан, так и рабов. Это, насколько мне известно, первый призыв к всеобщей грамотности.

Следующее реализовали скандинавы, сто лет спустя, поняв, что законы императора будут лучше соблюдаться, если в каждой деревне будет по крайней мере один грамотный человек — ведь так в народе будет меньше кривотолков. А это путь к сплочению империи. Однако сложность состояла в том, что технология записи была настолько плоха, что на обучение чтению тратилась уйма времени и сил.

Лично я считаю большой заслугой одно из нововведений Алкуина Йоркского, который вскоре после этого придумал вставлять пробелы между словами.

LXF: Почему нам в школе об этом не говорили?!

РА: Это же из ряда вещей, само собой разумеющихся: «Естественно, слова разделяются пробелами». Однако, особенно в латыни, куча фраз и неверных переводов возникли из-за того, что последовательность букв можно разделить минимум двумя способами, все время получая разный смысл — и какой из них считать верным? Потому-то рукописи и иллюстрировались. Чтобы визуально намекнуть, о чем там написано.

LXF: Рисунки появились раньше пробелов?

РА: И в каждом скриптории, ведь все переписывалось вручную, был свой стиль письма, то есть не было стандартного шрифта. Алкуин обращается туда и предлагает принять созданный им каролингский минускул, к которому ныне восходят все современные шрифты. Именно ему принадлежит идея стандартизации верхних и нижних выносных элементов букв, а затем последовало и добавление пробелов, и оформление слов как таковых.

Алкуин ввел целый ряд инноваций, благодаря которым люди смогли научиться читать. Последующие полвека называют Каролингским возрождением, мини-предвестником одноименной великой эпохи. Я утверждаю, что мы сейчас на пороге

высказывания «О, нужно всех обучить программированию», но уж больно это путаная и мудреная наука, ведь пробелов между словами, которые будут казаться очевидными и намного упростят это дело, у нас пока нет — нам нужен свой Алкуин Йоркский.

LXF: Значит, ждем пробелов между словами... тогда это точно не про Python, да?

РА [смеется]: Ну, вообще-то все это я говорил во время выступления на PyCon, доказывая, что добавление пробелов стало бы революцией. Так что аудитория Python, конечно, могла решить, что это про них. Но там было дело не только в пробелах. Возможно, это и про Python, но уверенно говорить об этом мы сможем только оглянувшись назад лет через 100...

LXF: Дело в концепциях, условных выражениях, стремлении найти решение проблем, которые до сих пор имеют место. Чего не-программистам не понять. Это развеивание мифа для огромного количества людей по всему миру, считающих, что компьютеры — это сплошное колдовство. Как с этим быть?

РА: Безусловно, это более концептуальный вид образования. Около года назад я решил написать об этом книгу. Думал привести тот тезис, о котором мы только что говорили — о Карле Великом и о рубеже, когда всеобщая грамотность может легко стать реальностью, и о том, что всеобщая компьютерная грамотность аналогична лингвистической, и о том, как нам этого добиться. Но в процессе работы моя мысль развивалась. Я наткнулся еще на кое-какой интересный материал, прочитав

который, я понял, что мой тезис не выдерживает критики!

LXF: А как же книга?

РА: Книга будет! Я лишь отодвину сроки, поскольку теперь мне нужно время подумать и подобрать новый тезис, который покажется убедительным мне самому и будет более продуктивным. По большому счету, это мысль о том, что для грамотности, со времен Карла Великого и за всю последующую историю ее преобразований, было два определяющих стимула, как и для образования, и что грамотность и образование всегда строились вокруг чтения. Последнее обусловлено двумя социальными мифами; первый — это христианский миф: чтобы понять Божье слово, следовало его прочесть. «Читать» значило «становиться лучше». Ради этого и следовало читать — это был своего рода социальный стимул. А раз от этого становятся лучше, стало быть, всем следует слушаться. И далее второй миф, актуальный для нашей колонизированной Америки, да и для всего мира в целом, о том, что настоящее демократическое общество, как и любое другое, должно строиться на просвещенности его граждан. Для нас это было важно, чтобы они смогли голосовать. Для недемократических обществ —

ОБ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ «В будущем станет понятно — нам нужен свой Алкуин Йоркский.»

поскольку так они будут знать законы и повиноваться им, не создавая себе неприятностей. Но чтобы быть просвещенными, граждане должны уметь читать, а следовательно, их необходимо »



учить. Та-да! Ведь это делает их более полноценными членами общества.

Чтение, по словам Деборы Брандт [Deborah Brandt — почетный профессор англистики Университета Висконсина] — это «на благо», а не «благо». А вот письмо — уже «благо». То, что вами написано — уже продукт, а потому его можно продать. Благодаря письму вы не совершенствуетесь... Таков миф. И если взглянуть на информатику 1980-х, в *Literate Programming* Дональда Кнута [Donald Knuth] читаем: «Да, нужно писать понятные программы, чтобы другие люди могли их читать». В основе этого — укорененное в культуре представление о том, что «чтение — это хорошо». Я же имею в виду нечто другое.

Второй частичкой пазла является тезис, над развитием которого я работаю сейчас. Изначально никто не писал и не читал просто потому, что этого еще никто не выдумал, или это служило только для ведения счетов. Никакого социального эффекта письмо не имело просто потому, что не выходило за пределы государственной казны. Потом Сократ, как мы знаем от Платона, считал письмо плохой идеей. Платон с Сократом согласен, но сам же пишет трактат «Государство», чтобы это обосновать.

LXF: Вот это размах: от 1980 до 400 лет до нашей эры.

РА [отсмеявшись]: После Аристотеля и Платона, читателей было мало, но на то были социальные причины; писателей тоже. Все меньше людей

читает, все меньше людей пишет. И так было, грубо говоря, до Гутенберга — и до печатного станка. Так называемая «рукописная культура». В рамках такой культуры отсутствуют понятия авторского права, оригинала, плагиата, поскольку при столь малом количестве читателей это неактуально. Главная идея в том, что самостоятельно создать текст полностью, с нуля, невозможно: это непосильный труд, посему вы просто компилируете его из других, уже готовых текстов, и понятия авторства как такового вообще не существует.

LXF: И эти тексты исправлялись по мере копирования?

РА: Намеренно или случайно, но да. Иногда случались ошибки при передаче звучания слов — читать об этом порой невероятно интересно. Иногда имела место антологизация, по принципу: «вот неплохая вещьца», а вот еще одна, почему бы их не соединить. Иногда первоисточник указывался, иногда нет — какое это имело значение?

LXF: Когда нельзя было поискать первоисточник в Google...

РА: Верно. Так и возникло представление об интертекстуальности. О текстах, порождающих тексты. Тексты с нуля не пишут. Таков был этос рукописи, и поскольку читателей не было, не было и коммерческого интереса в письме и чтении. А затем появилось книгопечатание и массовое чтение. Письмо оказалось одним из «благ», читать стало

нужно «на благо». Писателей было по-прежнему мало, поскольку для того, чтобы обеспечить пространство, доступность и окупаемость, требовалась техника. Но появилась коммерческая выгода. То, что вы пишете, может принести вам деньги, а если кто-то это «украдет»... тут возникают идеи об авторском праве, плагиате и авторстве, само понятие об авторстве. Кто там автор этого свитка, с чего бы ему писать свое имя, какое кому до этого дело? Вам не важно, им не важно — никому не важно.

LXF: Саму писателю это и в голову не придет.

РА: ...По моему мнению, сейчас мы вступаем в четвертую эру. Да, это эра Дейва Вайнера [Dave Win-ner], где есть блоги, SMS, Twitter. Мы все стали больше писать. Многие сегодня пишут больше, чем Диккенс написал за всю свою жизнь. Похуже, чем Диккенс, но больше, чем Диккенс! Речь об объеме. Говорят, такое сравнение неправомерно, ведь люди сегодня пишут, в основном, дурацкие шуточки и всякую ерунду, а Диккенс был гением. Но при тысячекратном увеличении объема, даже если в целом качество упадет на 50%, качественных вещей все равно станет больше. Уровень повышается.

Обращаясь вновь к глобальному соотношению читателей и писателей, сегодня оно вновь возвращается к [следует драматическая пауза] средневековым показателям. Было мало и мало, стало — много и много. Что же произойдет, когда это соотношение восстановится? Вот об этом я пока еще думаю, но отчасти это приведет к возвращению средневековых моделей; возможно, поэтому мы сейчас наблюдаем ослабление авторского права, и не только в сфере ПО, но и в сфере музыки и журналистики. Представления о том, что такое хорошо и что такое плохо, оказались подорваны не только в бизнесе, но в социально-этической плоскости, в целом. И с этим пытаются бороться. Однако есть еще одно изменение, касающееся уже напрямую кода, которое, как мне видится, не является простым повторением средневековой модели на другом уровне — о нем я узнал из Проекта Эйлер [Project Euler] (<http://projecteuler.net>).

Когда-то Проект Эйлер отпочковался от обучающего математического сайта и обрел большую популярность, представив 450 задач, изначально математических, исходя из специфики сайта, но из разряда тех, что решить можно только с помощью компьютера. Первая задача очень проста: какова сумма всех чисел меньше миллиона, кратных 3 и 5? Это можно посчитать при помощи ручки и бумаги, но нужно написать для этого программу, что существенно усложняет задачу. Лично меня зацепило то, что написано в разделе FAQ на этом сайте: «О, я нашел прекрасное решение в моем любимом языке, можно выложить его в Интернет?» И ответ: «Вы уже сами себе ответили». Вам знакомо это чувство гордости, когда вы нашли прекрасное решение — так не крадите его у других. Не выкладывайте свой код.

В данном случае, перед нами очень узкий пример, но если взять со времен Аристотеля, думаю, за тысячу лет найдутся еще аргументы, я имею



в виду ситуацию, когда кто-то говорит: «Публиковать исходный код — неправильно. Этически неправильно». Обобщая это с тем, о чем мы говорили в контексте грамотности, речь о письме. Заходя на Проект Эйлер, вы учитесь не посредством чтения, просто потому, что вы это умеете — это устаревший способ получать образование: вот прочитаю эти алгоритмы и поумнею — нет, нет, нет и нет. Учиться вам предстоит, самому се за решение, написав его, ведь компьютер, в контексте грамотности, выполняет разграничительную функцию: ну написали вы что-то, но как узнать, плохо или хорошо? Нужен отклик читателя, обратная связь.

Когда вы пишете программу, компьютер как раз дает вам такой отклик. Возможно, в таком виде многим это непонятно, но у нас здесь действительно свое средневековье. Компьютер подсказывает, является ли этот алгоритм более или менее эффективным, чем тот, который вы пробовали прежде, верный вы получили ответ или нет. Это и есть своего рода обратная связь, которую вы получаете, когда пишете программу и переписываете ее. Лучше становитесь именно вы, и именно при помощи письма. В этом случае, письмо — это «благо». А когда вы видите, как одно и то же всплывает и в Twitter, и в Facebook, культурный посыл таков: «Публиковать посты нужно, ведь письмо — это продукт». Стимул тот же, как ко-



мите человека, который абсолютно ничего не знает о компьютерах, хоть взрослого, хоть ребенка, и дайте ему программу, написанную на каком-нибудь из ваших любимых языков программирования: Python, Java, Smalltalk.

LXF: Perl?

РА: Да хоть бы и Perl, не важно. Просто готовую программу. Смогут ли они ее запустить? Или настроить

параметры, если это нужно? Или понять, нужна она им или нет, и получить нужный результат? Смогут ли они ее использовать, прочитать? Думаю, ответом почти всегда будет «нет». Отчасти это опять касается пробелов между словами, нечто из серии «почему же так трудно использовать уже написанную программу». Вероятно, с этого и стоит начать. Когда вы учите людей писать код, а они даже не представляют, что с ним потом делать, не ставите ли вы телегу впереди лошади? Возможно, начать стоило с того, чем так силен Open Source: «Заходите на сайт, выбираете, что именно хотите сделать, и, раз кто-то уже наверхняк создал для этого все инструменты, устанавливаете их и приступаете». Хотите, чтобы у вашей программы был свой блог — отправляетесь на blogger.com и получаете учетную записку, а если хотите мыслить как программист — получаете необходимый код, устанавливаете его и выполняете.

В реальности, программист этот самый код напишет сам, а потом установит и выполнит, но давайте пропустим этап написания и проверим, сможем ли мы сделать все остальное, и что, на самом деле, сложнее? И что нужно знать тем, кто не собирается становиться профессиональными программистами?

LXF: То есть что поможет им понять, как это все работает? Нужна ли им вся эта супер-архитектура, столь привлекательная для знатоков и для тех, кто хочет ими стать?

РА: Именно так. Согласен с вами на 100%. Понять, грубо говоря, как устроен автомобиль, чтобы знать, что он не летает. Просто уметь взять уже готовый код и посмотреть, насколько быстро он работает, для чего он нужен. Что-то в этом духе, в том числе и написание кода. Не уверен, что именно с этого нужно начинать, но у нас так и происходит.

LXF: Так принято, так учили тех, кто учит нас.

РА: Откуда все и идет, и это, по моему мнению, есть пережиток прошлого века... когда вы видите, что при обучении людей программированию речь идет о том, чтобы сделать их экономически конкурентоспособными. Никто не говорит, что чтение вас сделает таковым. Читать нужно для того, чтобы быть более полноценным членом общества. А если ваша отправная точка: «я не хочу быть программистом, когда вырасту. Я хочу стать журналистом. Я хочу стать танцором в балете»? Собственно, как-то так я и хотел начать свою книгу, поскольку мои жена и дети — люди творческие. С женой мы познакомились, когда она преподавала цирковые искусства: хождение по канату и прочее. Нужно ли ей было для этого уметь читать? Нет. Но почему этому все же следует учиться? Потому что вы должны знать, как вести себя в обществе, верно? Профессии здесь ни при чем. Чтение не делает вас конкурентоспособнее, и если цель в том, чтобы развеять мифы о компьютерах у обычных людей, дабы им легче жилось в мире, столь сильно зависящем от техники, нужно сделать так, чтобы они понимали принцип их действия, а не использовали в работе. Они станут частью повседневной жизни. Как? Я не знаю. **LXF**

ОБ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

«Оно опирается на письмо, а не на чтение, чем и отличается от грамотности.»

гда-то — учиться читать, а теперь — учиться программировать; но, если вдуматься, как мы учим программировать? Через написание собственных программ, а не через чтение чужих. Исходим ли мы при этом из этических или из практических соображений? Если бы существовал некий канонический код, стоило бы людям его читать? Я думаю, нет. Когда я развиваю эту мысль логически, у меня возникает когнитивный диссонанс, ведь при таких принципах обучения программированию открытый код, по сути, станет злом.

LXF: Но ведь это только на стадии обучения!

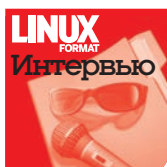
РА: И все же этот когнитивный диссонанс казался мне достаточно серьезным поводом, чтобы задуматься. Отправной точкой для меня, я извиняюсь за столь длинный ответ, стал вопрос о том, каково будущее обучения программированию? Похоже, что оно опирается в основном на письмо, а не на чтение, чем и отличается от другого вида грамотности.

LXF: Начинать надо сразу с написания кода?

РА: Это рассуждение в долгосрочной перспективе. В короткой же, я всегда считал, и не изменил своего мнения, что мы не должны с этого начинать, продолжая наш мысленный эксперимент — возь-

Виталий Липатов: Почти из мифологии

Игорь Штомпель расспросил его о трудностях развития, выпускаемых продуктах и отношениях с сообществом.



Виталий Липатов — генеральный директор компании Etersoft и один из ее основателей. Компания Etersoft известна решениями, которые позволяют запускать в Linux

необходимое для предприятий программное обеспечение, созданное для использования в операционной системе Windows.

Linux Format: Расскажите об истории компании Etersoft. Что привело к ее появлению?

Виталий Липатов: Дело в том, что это уже практически мифология; сложилась некая лубочная картинка о появлении компании.

Если просто, то мы с друзьями в начале 2000-х были вдохновлены тем, что открыли для себя Linux, вдохновлены примером компании ALT Linux, собравшей коллектив разработчиков — ALT Linux Team, в деятельности которой я стал принимать активное участие (собирал пакеты).

И мы решили, что нашими знаниями надо делиться, и всем внедрять Linux. Организовали компанию. Но оказалось, что Linux никому не нужен без бизнес-программ, в частности, 1С. Предприятие, поэтому нам пришлось переключиться на разработку Wine, запускать и внедрять решения 1С + Linux. Это было сложно и долго. Только в последние пару лет мы наконец смогли вернуться к изначальной теме и заняться внедрениями и заказной разработкой.

Поскольку у нас много своих продуктов на тему миграции с Windows на Linux, самостоятельные внедрения нам очень важны, ведь они являются опытом, непосредственно воспринимаемым разработчиками через наших сисадминов-внедренцев. Так мы пытаемся видеть, то ли мы делаем, удобны ли наши разработки. Конечно, обычно мы занимаемся только удаленными внедрениями, когда личное присутствие не требуется, то есть

это обычно совместная работа с сисадминами заказчика.

LXF: С какими трудностями вам пришлось столкнуться в развитии компании? Какие проблемы актуальны на сегодняшний день?

ВЛ: Мне кажется, что основная трудность — это неумение развивать компанию. Если подробнее, это и подход к получению сотрудниками необходимых навыков, и дисциплина в компании, и финансовая сторона — у нас не было какого-то инвестора, и дела шли очень слабо. В ставшей классической модели разработки ПО сначала нужно долго трудиться, создавая продукт, а потом продавать его массовому потребителю по фиксированной цене. Конечно, требуются инвестиции, чтобы дожить до коммерческой отдачи.

Использование СПО дает возможность более плавного старта: можно сразу начинать продавать свои доработки. К сожалению, на подход «мы про-

«Чувствуем обязательство возвращать наработки сообществу.»

даем услуги по доработке свободного ПО» мы выходили очень долго. Во многом повлияло и то, что у предприятий сложился стереотипный подход к покупке ПО как к покупке коробки: оплатили, распаковали, используем. Все остальное выходит за рамки привычной схемы и непонятно бухгалтерии.

Следующая большая трудность — выбранный сегмент рынка. Заниматься разработкой под Linux в условиях, когда рынка практически нет, особенно в области прикладного (офисного) ПО, крайне неэффективно. Это замкнутый круг, который мы различными усилиями старались разорвать.

Из проблем я бы отметил кадровую. Не касаясь уровня подготовки в ВУЗах (думаю, кто все же хотел научиться, тот научился), скажу, что рынок программистов сильно перекошен, в частности, присутствием в России центров разработки иностранных компаний. Не знаю, кто придумал этому радоваться.

Во-первых, это беспощадный вывоз произведенного в России интеллектуального продукта. Во-вторых, программистам там платят завышенные по нашим меркам зарплаты, и, немного завидуя тем, кто их получает, хочу отметить, что иностранные компании сильно экономят на зарплате, при этом легко привлекая к себе разработчиков из отечественных компаний. Большой перекос в зарплате при равном уровне требований увеличивает текучку. С учетом того, что каждого специалиста приходится «растить» на месте год или два, в силу специфики нашей деятельности, это сильно мешает: как только специалист вырастает, его сразу ждут в соседнем здании.

Также сильно препятствует то, что мы живем «от проекта»: идет проект, есть по нему финансирование — работаем; нет проекта — исследованиями и новациями не особо занимаешься, нет такого финансового запаса в компании. Это тормозит рост компании, а еще мешает с полной отдачей возвращать наши наработки сообществу: используя свободный код, мы чувствуем за собой такое обязательство.

LXF: Какие отношения у компании Etersoft с российским Open Source-сообществом?

ВЛ: Ведутся, конечно, споры, существует ли такое сообщество, и какое из сообществ является правильным, и кто может говорить от его имени.

Мне кажется, что на самом деле сообщество достаточно малое (даже если брать в расчет не только разработчиков, но и пользователей-



энтузиастов и системных администраторов), и мы в любом случае со всеми общаемся, так или иначе. Как ответить на этот вопрос... Мы часть этого сообщества, я так считаю. И как сама компания Etersoft, так и отдельные сотрудники.

С коллегами и единомышленниками всегда общаемся, стараясь поставить общие цели и решение технических вопросов выше корпоративных интересов.

LXF: **Насколько большой штат у компании Etersoft?**

ВЛ: Штат небольшой, в пределах 20 человек. Бывают моменты, когда мы «растем под проект». Также привлекаем удаленных и внештатных разработчиков при необходимости. Мы существуем уже почти 10 лет (в начале 2014 года будем отмечать эту круглую дату), и круг общения у нас достаточно широк.

LXF: **Решения Etersoft основаны на Wine. А в чем заключаются их основные отличия?**

ВЛ: Линейка продуктов WINE@Etersoft состоит из 5-ти вариантов: Local, Network, SQL, CAD, School.

Во многом отличия касаются только направленности технической поддержки: Local — самая базовая версия, в Network мы поддерживаем работу программ, совместно работающих с данными по локальной сети, в SQL — работу с СУБД (такими как *MS SQL*, *PostgreSQL*). CAD-версия заточена на работу систем проектирования (например, продукта *КОМПАС* компании АСКОН). School — это на самом деле просто такая лицензия. По сути, это тот же Network, который мы бесплатно лицензировали для образовательных учреждений, чтобы как-то поддержать их инициативу перехода на Linux в условиях, когда государство забыло про свои планы перехода на СПО.

LXF: **Среди различных версий WINE@Etersoft есть версия Public, которая доступна свободно. Не могли бы вы рассказать о ней подробнее? В чем заключаются ее особенности и отличия от других версий?**

ВЛ: Версия Public включает те наши разработки, которые мы не смогли пока что передать в основную ветку *Wine*, но при этом не являющиеся хаками, которые чинят одно и ломают другое. Также она нацелена на те случаи, когда обязательно нужно внедрить *Wine* именно со свободной лицензией.

LXF: **Есть ли у компании Etersoft полностью свободные решения?**

ВЛ: Мы делаем свою сборку СУБД *PostgreSQL* под все платформы, с добавлением патчей *1C*, некоторых наших исправлений. Распространяется она под свободной лицензией, без каких-либо ограничений. К сожалению, до сих пор к нам

в поддержку обращаются люди и просят бесплатно помочь им с настройками или внедрением нашего свободного решения («мы тут скачали с вашего FTP пакеты, а они не работают»). Конечно, именно для этого мы предоставляем коммерческие версии, чтобы у нас с наших клиентов возникли взаимные обязательства и мы могли обеспечить поддержку.

Также на базе *NX*, *freemx* и *openmx* мы делаем продукт *RX@Etersoft* для организации терминальных серверов Linux- и Windows-приложений.

Есть решение, которое используется для *RX* и *Wine*: это модуль ядра *etercifs*. По сути, мы активно участвуем в разработке модуля *cifs* в ядре Linux. Но некоторые наработки, касающиеся, например, поддержки NT-семантики блокирования файлов в Linux-ядре, до сих пор у нас не приняты, поэтому мы их предлагаем в нашем модуле *etercifs*. Стоит отметить его особенность: *etercifs* собирается из исходников, и пакет устроен таким образом, что собирается на любой версии ядра и на любом дистрибутиве, благодаря набору модификаций *etercifs*, соответствующих каждой версии ядра.

Есть подзаброшенное решение: *UniOffice*. Оно разработано для Windows, и позволяет программам, рассчитанным на тесную интеграцию с *Microsoft Word*, работать вместо этого с *OpenOffice.org*. С программной точки зрения *UniOffice* эмулирует API *MS Word*, «притворяется» этой программой.

LXF: Расскажите о процессе включения наработки компании Etersoft в проект Wine.

ВЛ: Вообще добиться принятия изменений достаточно сложно. В общем случае получается два этапа:

1) Нужно написать тест, который доказывает наличие проблемы в текущем коде *Wine*. Он будет запущен на множестве тестовых машин с разными системами Windows, и должен на них успешно пройти, а для *Wine* показать неуспех.

2) Нужно предложить исправление проблемы, так, чтобы и для *Wine* этот тест проходил.

Это непросто и затрудняет продвижение вперед, но зато все реже бывает, что исправление одного участка ломает другой. К тому же с каждым исправлением код *Wine* покрывается все новыми

и новыми тестами, выявляя нюансы в поведении функций WinAPI.

LXF: Расскажите о вкладе компании в развитие ядра Linux.

ВЛ: Да не такой уж этот вклад и большой. Если не считать действительно существенного вклада в разработку модуля *cifs* (активная разработка которого также велась Стивом Френчем [Steve French] из IBM, а также в рамках Google Summer of Code), остальное — это небольшие багфиксы для NFS и FUSE.

У нас есть небольшие проекты, которые сильно помогают нам в работе: сборочная система *Korinf*, универсальный менеджер пакетов *EPM*, универсальный архиватор *ERC*, а также *gitum* — надстройка над Git, позволяющая без больших усилий вести форки проектов.

LXF: В чем особенности упомянутых вами проектов Korinf, EPM, ERC и gitum? Что послужило толчком к их появлению?

ВЛ: Решения возникли из сочетания разных направлений деятельности в компании: сборка наших продуктов в пакет для конкретных систем и услуги по внедрению наших продуктов.

Для наших программных продуктов мы при-

«Мы всегда имеем дело со множеством различных систем.»

держиваемся подхода, что они должны быть собраны под каждую систему персонально. Для автоматизации этого процесса нами была постепенно разработана сборочная среда, получившая название *Korinf*. Она позволяет из одного исходного (*src.rpm*) пакета, подготовленного в соответствии с требованиями для размещения в репозиториях ALT Linux, собрать двоичные пакеты для любой системы — от FreeBSD до Fedora, от Gentoo до Ubuntu.

При сборке пакетов в *Korinf* приходится производить манипуляции с пакетами: устанавливать необходимые пакеты для сборки, проверять зависимости, устанавливать собранные двоичные пакеты и т. д.

При администрировании системы решаются те же задачи: установка и удаление пакетов, обновление системы, поиск пакетов и пр.

Поскольку мы всегда имеем дело со множеством различных систем, то оказалось удобным для любой системы иметь одинаковый интерфейс средства управления пакетами, вне зависимости от конкретной системы и применяемого в ней менеджера пакетов. Это упрощает жизнь человеку (не нужно помнить все команды — попробуйте, например, запомнить, какую надо выполнить команду, чтобы определить, к какому пакету принадлежит тот или иной файл), и упрощает написание скриптов. По сути *EPM* возник за пару дней — много необходимого кода уже было в *Korinf*, мы просто его отделили в отдельный проект.

Таким образом, *EPM* — это просто оболочка, скрывающая детали использования конкретного менеджера пакетов. Он имеет интерфейс командной строки с множеством синонимов для команд, так что пользователи Ubuntu, Fedora или Mandriva могут вводить привычные им команды. Т.е. *EPM* поддерживает не только разные пакетные менеджеры, но и разные навыки пользователей.

Например, для установки пакета можно выполнить

```
epm -i <название пакета>
```

или **epm install**, или **epmi**.

EPM можно взять уже собранный для вашей системы на <http://wiki.etersoft.ru/EPM> или собрать в свой дистрибутив самостоятельно: <https://github.com/vitlav/eeepm>.

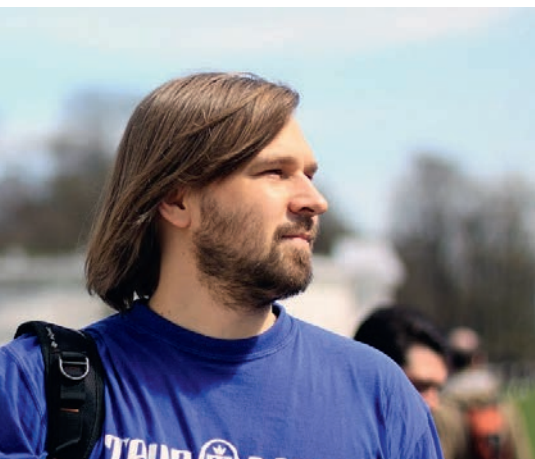
Оболочка для архиватора *ERC* следует тому же стилю, предоставляя общий интерфейс для архивов любого типа. Основные свои функции она реализует через *patool*.

Как работает *gitum*, нам всегда сложно объяснить. Попробую просто рассказать, какую задачу он призван решить. Представим, что есть проект, имеющий публичный git-репозиторий. Если вы участвуете в его разработке, то иногда передадите ваши коммиты, и их включают в общее дерево, которое является достаточно стройным. Но иногда бывают задачи, когда к такому проекту у вас имеется много изменений, которые по каким-то причинам у вас не приняты. Тогда вы создаете свой репозиторий, размещая туда свои добавления. Что происходит, когда вам нужно обновить

ваш репозиторий до состояния апстрима? Либо вы делаете **merge** (при этом ваши изменения остаются где-то далеко в истории репозитория, и постепенно размазываются по **merge**-коммитам из-за возникающих конфликтов при обновлении), либо обновление выполняется с **--rebase** (при этом ваши изменения всегда остаются поверх всего кода из апстрима — это очень наглядно и удобно, но из-за **rebase** теряется история изменений). *gitum* решает эту проблему, ведя несколько веток с кодом и осуществляя обновление из upstream-репозитория таким образом, что у вас есть и ветка, где выполнен **rebase** (ваши коммиты на виду), и ветка, где ведется история изменений (и апстрима, и ваших), и ветка, где ваши изменения доступны в виде отдельных файлов. Таким образом достигается возможность легкой синхронизации с upstream, не теряя при этом изменений и не создавая **merge**-коммитов, которые мы в данной ситуации считаем большим злом. Подобное слежение за апстримом может быть полезно и при сопровождении пакетов в репозиториях дистрибутивов (как известно, многие пакеты в репозиториях обрастают патчами, адаптирующими их к политике дистрибутива, чинящими сборку и пр.).

LXF: Что является препятствием для включения реализации поддержки NT-семантики блокирования файлов от Etersoft в ядро Linux?

ВЛ: Вариантов реализации было несколько, наше первоначальное предложение несколько раз





корректировалось. Нам потребовалось показать, как эта функциональность будет применена в файловых системах NFS и CIFS. Было много внимания уделено тому, чтобы ничего не поменялось для существующих программ. В общем-то это все трудности, но не препятствия. Последний из вариантов обсуждался здесь: <https://lkml.org/lkml/2013/4/26/242>.

Мы надеемся уже в начале года порадоваться успешному принятию нашей разработки.

LXF: Обеспечение поддержки каких программ в решениях Etersoft планируется на текущем этапе развития компании?

ВЛ: Существует большой пробел в средствах использования цифровой подписи. Не секрет, что большинство сайтов, где можно что-то удостоверить с ЭЦП, работают только в *Internet Explorer*, причем определенной версии, и требуют Windows-программ для функционирования. У нас есть планы поработать в этом направлении, чтобы изменить ситуацию и предоставить пользователям других браузеров и операционных систем более равные возможности. Сейчас есть проблемы и в работе с банками, и при участии в конкурсах и торгах, и при сдаче налоговой отчетности.

LXF: Планируется ли выпуск новых решений (имеется в виду, не новых версий, а расширение линейки программных продуктов)?

ВЛ: Пока я могу только сказать, что мы все же стали переходить от модели распространения лицензий на ПО к оказанию услуг по разработке и доработке ПО. Возможно, что косвенно из этого появятся свободные продукты, которые мы будем поддерживать. Мы понимаем, что пользователю важно и удобно получить весь необходимый набор ПО в комплексе, а не платить за каждую программу, продлевая различные лицензии. Одним из вариантов является поставка преднастроенной, готовой к работе виртуальной среды.

Также существует тенденция по «уходу в облака» — в тот момент, когда мы почти дождались появления Open Source на рабочих местах, эти рабочие места переместились в проприетарные облака: вы подключаетесь браузером к удаленному сервису в неизвестной стране, и понятия не имеете о лицензиях и программном коде на той стороне. Я не считаю это верным путем, но ориентироваться на эту тенденцию приходится.

LXF: Какие программное обеспечение вы используете в своей работе (дистрибутивы, рабочие среды, инструменты разработки, почтовые клиенты и т.п.)? Есть ли предпочтения для аппаратной составляющей?

ВЛ: В качестве рабочих систем мы используем различные версии ALT Linux. Нам это удобно, потому что мы сами участвуем в его разработке и можем легко повлиять на систему. К тому же так

мы поддерживаем отечественного разработчика, что нам представляется важным.

С почтой мы работаем обычно через web-интерфейс *RoundCube*. Но у каждого свои предпочтения: некоторые используют *Kmail* (я сам им лет 10 пользовался, пока он совсем не испортился в KDE 4), некоторые ставят *Thunderbird*.

Основной инструмент разработки — это *mc*. Еще мы используем *kdevelop* (для небольших проектов) или *QtDesigner* (для проектов на *Qt*), а также *kscope* (для больших проектов — других средств, которые способны проиндексировать ядро Linux или *Wine*, мы не нашли).

Ну, конечно, *Git* как систему контроля версий. *Bugzilla* для отслеживания ошибок. *Request Tracker* для отслеживания заявок клиентов.

Графическая среда обычно *Xfce*, как нересурсоемкая. Активно используется виртуализация на базе OpenVZ, а также стенд для виртуальных машин на базе *VirtualBox*. Аппаратные предпочтения обычно выглядели примерно так, продолжая название дистрибутива (ALT Linux): видеокарты ATI и процессоры AMD.

LXF: Используете ли социальные сети и/или интернет-сервисы в своей работе?

ВЛ: Социальные сети... На работе используем, а в работе — вряд ли... Интернет-сервисы используем, но, как правило, свои. У нас есть принцип не завязываться на сторонние web-сервисы. **LXF**



Джоно Бэкон листает пыльные страницы летописи и выделяет ключевой компонент в машине открытого кода: проект Qt.

Когда Голливуд решит снять блокбастер по истории Linux, в нем будет немало персонажей, играющих важные роли: ядро, разные среды рабочего стола и программы загрузки, но есть один персонаж, который выдержал испытание временем, претерпев куда больше невзгод, чем он заслуживает, и это — проект Qt.

Qt — это графический набор инструментов и платформа разработки приложений, у которой долгая и интересная история и которая, несмотря на все беды и испытания, на данный момент стала одним из самых важных элементов многих проектов с открытым кодом.

Сегодня мы пройдем по Тропе Памяти, чтобы ознакомиться с удивительно долгой историей Qt и узнать, каково ее состояние на сегодняшний день и что день грядущий ей готовит.

Qt, что произносится «кьют», а не «кью-ти», появилась на свет в 1991 году, когда Хаавард Норд [Haavard Nord] и Эйрик Хамбе-Энг [Eirik Chambe-Eng] начали работу над этим набором инструментов. По мере того, как Норд и Хамбе-Энг продолжали работу над своим новым проектом, они создали для него Quasar Technologies, а через три года основали Troll Tech (который в конечном итоге стал Trolltech).

Идем по Тропе Памяти

В те определяющие для Qt дни тандем работал над созданием подборки графических компонентов набора инструментов. Эти компоненты состояли из кнопок, полос прокрутки, списков, комбо-блоков, текстовых блоков, просмотра значков и прочих знакомых всем графических блоков, необходимых для создания приложений.

Qt работала на популярной и вездесущей X Window System (тогда она выступала под названием XFree86) и была написана на мощном и эффективном, кросс-платформенном языке программирования C++.

C++ был выгоден многим разработчикам. В Linux уже присутствовал мощный компилятор C++, и C++ предоставлял также возможности объектно-ориентированного программирования (ООП), которые отлично подходили для написания графических приложений в виде множества отдельных частей, обязанных работать вместе. Как таковая, Qt предоставляла набор повторно используемых объектов C++, а каждый виджет предоставлял богатый API с определенными функциями и поведением (именуемыми сигналами), и даже включал удобные классы для облегчения сложностей в C++ (например, объект обработки

строки QString, предоставлявший более простой способ работы со структурами данных в приложении).

Как только Норд и Хамбе-Энг выпустили публичный релиз своего нового набора инструментов, тот привлек внимание ряда программистов. К тому времени, в середине — конце девяностых, уровень функциональности, имеющийся в Qt, в основном отсутствовал в большинстве других наборов инструментов, таких, как Motif, свободный эквивалент Lesstif, Athena и новичок — GTK. Один из аспектов Qt также заинтересовал многих наблюдателей: внешний вид и ощущение от работы с Qt можно было настроить просто с помощью тем, и в результате приложения Qt выглядели гораздо лучше, чем громоздкие альтернативы Motif/Lesstif. Все, кто помнит первую версию Netscape для Linux, точно знают, о чем я говорю.

Одним из разработчиков, с восторгом принявших Qt, был Маттиас Эттрих [Matthias Ettrich], который тогда был студентом Eberhard Karls University в Тюбингене, Германия. Эттрих был страстным поклонником Unix и Linux, но не имел иллюзий насчет рабочих столов для этих ОС. Его огорчало, что приложения так разномасштабны, и ему были нужны гармоничные рабочий стол и работа с приложениями. Эттрих также чувствовал, что приложения того времени слишком сложны в использовании, и он мечтал о простом и удобном рабочем столе, понятном даже начинающим пользователям. Так и появился проект KDE.

В своем первом обращении к общественности (приведенном ниже) о своем новом проекте "Kool Desktop Environment" в группе Usenet de.comp.os.linux.misc он поделился своим желанием использовать Qt:

«Вот уже несколько недель свободно и бесплатно доступен исходник действительно превосходной новой библиотеки виджетов для разработки свободного ПО. Загляните на www.troll.no.

Программа называется Qt и является настоящей революцией в программировании X. Это почти полная библиотека виджетов полностью на C++, которая реализует слегка улучшенный внешний вид и способ работы Motif или Windows 95, который можно включить при запуске.

Тот факт, что она делается компанией (Troll Tech), по моему мнению, является огромным преимуществом. У нас есть ресурсы, превосходная библиотека, а у них есть бета-тестеры. И они посвящают ВСЕ СВОЕ ВРЕМЯ улучшению библиотеки. Они обеспечивают отличную поддержку. А значит, Qt также представляет интерес для коммерческих приложений. Реальная альтернатива ужасной Motif :) Но самое большое преимущество Qt в том, как она запрограммирована. Это действительно очень простая в использовании библиотека C++.

Qt также портируется, пока в Windows 95/NT, но не стоит об этом волноваться. Очень легко использовать специфические вещи UNIX/X в программировании, поэтому это портирование в NT вряд ли возможно :-».

По мере роста сообществ Linux и свободного ПО, Qt быстро становилась ключевым компонентом в истории роста, замахнувшись на то, чтобы потеснить крутых парней в проприетарном ПО. Однако, к сожалению, Qt понравилась далеко не всем, поскольку ее лицензия вызывала некие сомнения.

Проблемы с лицензированием

При своем появлении Qt вышла под лицензией FreeQt, дойдя под ней до версии 1.45. Для некоторых членов сообщества СПО лицензия FreeQt была весьма спорным пунктом, поскольку эта лицензия не разрешала модификаций исходного кода Qt, то есть основной свободы, гарантированной GNU Public License. Те же критики нередко восхваляли технические способности Qt, однако проблема с лицензией в их глазах являлась этическим барьером.

Пытаясь решить эту проблему, версия 2.0 Qt вышла под другой лицензией, Q Public License (QPL). И хотя QPL предоставляла



➤ Ларс Кнолл (справа) являет собой значимую фигуру в управлении как Qt, так и KDE.

многие из основных свобод в лицензии СПО и многие считали ее дееспособной лицензией свободного ПО, Free Software Foundation [Фонд Свободного ПО], который и создал GPL, не считал QPL совместимой с GPL. Qt снова оказалась отброшена назад.

Это обострило отношения между Trolltech и проектом KDE. Проект KDE начал ощущать возрастающее давление, вынуждающее его отказаться от Qt в силу проблем с лицензией.

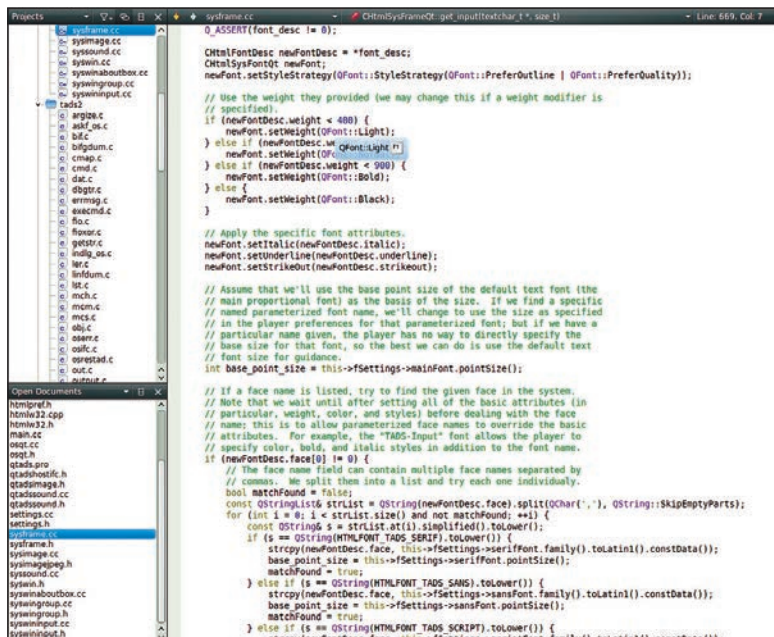
На минуту спустившись с кафедры LXF, автор живо вспоминает то время, поскольку он активно участвовал в проекте KDE и испытал всю тяжесть нападков на Qt, являясь в тот момент представителем KDE в Великобритании. Не проходило ни дня без стычек и споров по этому вопросу, мешавших и команде KDE, и команде Qt сфокусировать все свое внимание на создании отличной программы. Многих эта проблема весьма удручала; свободу ценят все, но люди чувствовали, что такие дебаты являются междоусобицей, тормозящей работу внутри Linux. Автор разделял эту точку зрения, хотя и поддерживал призывы к более свободному Qt, но также и выгодному и для свободного ПО, и для разработчиков коммерческих приложений.

В те времена предприняли две попытки решить эту проблему. Первой был довольно амбициозный проект Harmony, который пытался создать совместимый по API и полностью свободный эквивалент Qt. А вторая заключалась в нелегко давшемся дипломатическом соглашении проектов KDE и Qt, что Qt не подпадет под более ограничительную лицензию, даже если некая компания приобретет Trolltech. В результате появился KDE Free Qt Foundation, гарантировавший, что Qt будет выходить под

Управление и сообщество

Проект Qt вначале основывался на весьма интересном наборе идей и целей от своих основателей, Хааварда Норда и Эйрика Хамбе-Энга, и далее превращался во все более открытый и совместно управляемый проект. Qt работает как типичный проект с открытым кодом. Разработка ведется открыто, через списки рассылки, каналы IRC, программы отслеживания ошибок, страницы wiki и форумы. Присоединиться к проекту может каждый,

и многие коммерческие организации активно ему помогают. Среди них — Digia, шведская подрядная компания KDAB, RIM (поддержка Blackberry и QNX), Intel (поддержка Wayland) и AudioCodes (поддержка ClearCase в Qt Creator). Qt также является проектом с открытым руководством, который работает под знаменем Qt Project. Очень важно, что это руководство всесторонне поощряет сотрудничество и основано на модели открытого руководства WebKit.



Qt Creator дает все необходимое для создания приложений Qt в одной посуде.

BSD-подобной лицензией, при условии, что ни одна свободная/с открытым кодом версия Qt не выйдет в течение 12 месяцев.

К счастью, когда мы вошли в новое тысячелетие и популярность KDE как рабочего стола для систем Unix и Linux росла, в Trolltech приняли решение выпустить Qt под лицензией GPL v2.0, положив конец разногласиям — по крайней мере, по части зависимости проекта KDE от Qt.

Но одна проблема осталась: популярнейшим атрибутом Qt была ее способность к работе на разных платформах, а Windows-порт Qt пока что не был доступен под лицензией GPL v2.0. Хотя многих из участников споров о лицензиях радовало то, что Trolltech предоставил порт Linux свободно и делает деньги на лицензиях Windows, были и такие, кого по-прежнему раздражало, что переносимости Qt мешают лицензионные ограничения.

В 2002 году члены команды KDE по проекту Cygwin предприняли попытку решить эту проблему и начали портировать лицензированную под GPL Qt для кодовой базы X Window System в Windows. Этот амбициозный проект достиг весьма достойных результатов, хотя его и отправили на полку, когда Trolltech выпустил Qt4 для Windows под лицензией GPL в 2005 году. Теперь вся платформа Qt была доступна под лицензией свободного ПО, и можно было легко создавать и распространять приложения с открытым кодом.

Коммерческий прорыв

Традиционно у проекта Qt всегда был явный лидер и коммерческий спонсор в лице Trolltech. Компания была основана создателями Qt и продолжала принимать на работу многих известных разработчиков Qt и инженеров сообщества. Для тех, кто пристально следил за проектом KDE, Trolltech стал настоящим пылесосом, вытягивавшим самых талантливых, чтобы они работали

на благо Qt и, таким образом, предоставляли лучший набор инструментов для создания KDE.

Хотя традиционно Trolltech уделял внимание рынку рабочих столов, совместимых с Linux, Windows и Mac, приход ядра Linux, способного работать на мобильных устройствах, становился все более и более заманчивым. Эту революцию по большей части подстрекуло устройство iPaq от HP, на котором работали Linux и подмножество приложений. Хотя технически приложения работали, многие из них выглядели приложениями настольного ПК, работающими на маленьком экране. Нужен был набор графических инструментов, оптимизированных для мобильных экранов... хм, интересно, кто бы мог его предложить?

Задолго до захвата господства на рынке мобильных устройств системами iOS и Android, Trolltech разглядел возможности Qt на мобильных телефонах и начал работать над мобильным Qt. В результате появился Qt Embedded, мощный дистрибутив Qt для написания мобильных приложений. По сути, Qt Embedded был таким же, как любой другой порт Qt, но работал быстрее на маломощном оборудовании, был оптимизирован для тачскрина и предоставлял функции уровня устройства. Это тоже был весьма достойный старт.

Сделки по устройству

Важной ступенью в этой работе была сделка, заключенная между Trolltech и Sharp по выпуску работающего на Linux мобильного устройства под названием Sharp Zaurus. Это был не телефон и не планшет, а PDA (тогда были популярны устройства Palm и Handspring), зато и одно из первых устройств в мире, которое изначально работало на Linux, и однозначно первым, работающим на Qt.

Несмотря на энтузиазм и интерес разработчиков (и автора), Zaurus стал эпическим провалом. Ограниченные спецификации и продажи, а также сложности в использовании приложений и возможности поделиться ими ограничивали шансы этого устройства, однако шансы для Qt в мобильной среде никуда не делись.

Это не осталось незамеченным в Nokia, которая в январе 2008 года объявила о своем приобретении Trolltech. В тот момент положение Nokia в мире мобильных устройств было далеко не идеальным, поскольку господство на рынке захватила Apple, а Android подтолкнул новинки гигантов по производству оборудования, HTC и Samsung. Nokia срочно нужно было что-то делать, и они выбрали Qt в качестве средства, на котором они построят мощную стратегию разработки.

К сожалению, все пошло не так, как ожидалось. И хотя Nokia метнула на рынок некоторые из своих сказочных «Интернет-планшетов», например, N900, в конечном итоге их постигла судьба Sharp Zaurus — они не нашли своего потребителя и страдали от ограничений по оборудованию и программам.

Для команды Qt и ее поклонников последовали напряженные годы. Nokia всаживала немалые средства, чтобы увидеть продажи своих продуктов, но слабый коммерческий успех привлек Nokia к затруднениям. И в результате глава Nokia Стивен Элоп (Stephen Elop) объявил о стратегическом партнерстве между Nokia и Microsoft.

Напишите свое первое приложение Qt

Решив создать собственное приложение Qt, предварительно проверьте все ресурсы, доступные в Сети, и только потом приступайте к кодированию:

- » Документация — <http://qt-project.org/doc>
- » Форумы поддержки — <http://qt-project.org/forums>
- » Региональные группы — <http://qt-project.org/groups>

Если хотите внести свою лепту в многоплатформенную стратегию Ubuntu, можете также написать конвергентное приложение QML (QML — часть Qt), способное работать на ПК, телефонах и планшетах. Этот метод как основную технологию использует QML плюс несколько спецкомпонентов Ubuntu. Более подробная информация по разработке — на <http://developer.ubuntu.com>.

Автор отлично помнит времена, когда его почтовый ящик ломился от писем сотрудников Nokia, с запросами о переходе на работу в Canonical. Времена были тяжелые, и многих тревожило будущее Qt. Казалось очевидным, что новая стратегия Nokia Microsoft не планирует поддержку Qt.

И дело не только в этом: многих жестких приверженцев открытого кода шокировала сделка с Microsoft. К счастью, напряжение ослабло в августе 2012 года, когда Digia, консалтинговая компания и долгосрочный инвестор в проект Qt, объявила о приобретении Qt у Nokia. Это означало, что изначальная команда разработчиков станет сотрудниками Digia.

Новая эра возможностей

Для многих история Qt казалась нисходящим движением по spirali до 2012 года, по причине сложных взаимоотношений с Nokia, малых финансовых вливаний в проект и спада числа разработчиков открытого кода, создававших приложения на Qt. Большинство дистрибутивов Linux начали также предлагать рабочие столы и средства разработки на базе GTK.

Однако с 2012 года у Qt открылось нечто вроде второго дыхания. Сейчас у проекта есть коммерческая поддержка в Digia, надежном и давнем друге проекта Qt, и ведется солидная разработка его основных функций и возможностей. (см. *Что за штука... Qt5*, стр. 54). В результате все больше организаций и проектов обращают свои взоры на Qt. Среди примеров — Blackberry, Jolla и Ubuntu. Ubuntu построил на Qt всю свою стратегию конвергенции, которая включает не только Unity (написанную на Qt), но работает на настольных ПК, телефонах, планшетах и ТВ. Разработка приложений Ubuntu тоже ведется на Qt. Фактически, скачав Ubuntu SDK, вы увидите, что SDK полностью работает из Qt Creator, ключевой части проекта Qt.

Все больше разработчиков, и открытого кода, и коммерческих программ, используют Qt как мощную платформу для создания своих приложений. Среди них такие организации и приложения, как Autodesk Maya, Mathematica, Европейское космическое агентство [European Space Agency], Dreamworks, DraftSight, Google Earth, HP Virtual Rooms, Lucasfilm, The Foundry's, Panasonic, Philips, Samsung, Siemens, Skype, Nuke, VirtualBox, VLC media player, Volvo и Walt Disney Animation Studios.

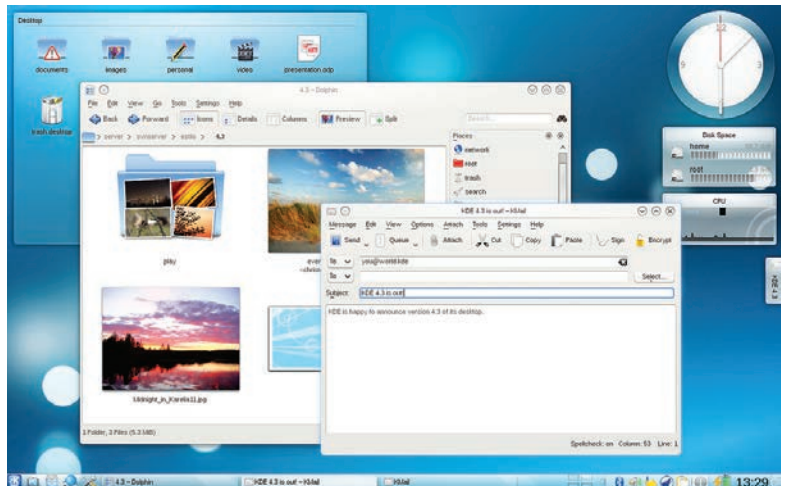
Технология

Главная причина этого роста в том, что, хотя история Qt знавала солидные потрясения, качество, целеустремленность и новаторство проекта всегда оставались стабильными, и сохранялась высокая степень мотивации. Qt долго выигрывала благодаря исключительному руководству, и Ларс Кнолл [Lars Knoll] является одним из незаменимых людей у штурвала. Ларс всегда сохранял целеустремленность проекта, его амбиции и работу над решением проблем, возникавших за все эти годы.

Первые несколько версий Qt были предназначены только для X Window Systems и Microsoft Windows, причем для Windows были доступны под проприетарной лицензией. С выходом Qt3, ближе к концу 2001 года, к числу поддерживаемых платформ добавились Mac OS X.

С тех пор Qt стала поддерживать Android, iOS, встроенные платформы, Mir, Wayland, QNX/Blackberry 10, vxWorks и Windows CE. Неофициально различные сообщества также портировали Qt в OpenSolaris, Haiku, OS/2, webOS, Amazon Kindle DX и AmigaOS. Nokia также предоставила порт Symbian, но он уже не поддерживается. Качество кодовой базы позволило Qt распространиться на разные платформы.

Изначально Qt появилась как простая подборка графических компонентов с несколькими удобными классами, которые все можно было использовать из C++. Qt предоставила эти основные классы и их MetaObject Compiler, который интерпретировал



настраиваемые макросы C++ в Qt и генерировал C++ для обеспечения функций, недоступных в C++ на тот момент, например, сигналов и слотов, интроспекции и асинхронных вызовов функций.

На заре существования Qt каждый новый релиз просто добавлялся в эту библиотеку графических компонентов и классов. Когда эта подборка достигла солидного уровня функций, проект начал все больше и больше уделять внимание инструментам в Qt.

Инструменты Qt

Одним из первых важных вкладов стал Qt Designer — графический инструмент, который ускориł и упростил раскладку пользовательского интерфейса. Шло время, и все больше подобных инструментов становились частью основного дистрибутива Qt, например, Qt Linguist, Tulip, Arthur, Scribe, MainWindow, QML и QSA.

Многие из этих инструментов в итоге слились в Qt Creator, полноценной IDE с интегрированными дизайнером, отладчиком ошибок, контролем кода и прочими функциями. Qt Creator был ключевым моментом в истории Qt: конкурирующие среды разработки приложений (такие, как Apple с XCode и Google с Android/Eclipse) строили целостную конструкцию IDE/Platform/Debugger, и проект Qt стремился создать нечто столь же привлекательное. Qt Creator сплотил отдельные компоненты в стабильную платформу.

Как видите, временами история проекта Qt была весьма простой, но мечта его создателей и многих разработчиков, которые помогли ее осуществить, о простом, мощном и переносимом наборе инструментов оставалась неизменной. Если возникали проблемы, проект Qt прислушивался, а если появлялись возможности, он их использовал.

Сейчас Qt пребывает в весьма неплохой ситуации, и его будущее выглядит позитивным и обеспеченным. Солидные организации используют решения и инструменты разработки Qt, Digia стала влиятельным коммерческим учреждением, и Qt продолжает набирать силу. Лично я сгораю от любопытства узнать, какой будет следующая глава в книге истории Qt! **LXF**

► Популярная среда KDE создана с помощью Qt и QML.

Узнайте больше

Если вы хотите узнать больше о спонсоре Qt, Digia, проекте Qt и о том, как связаться с сообществом Qt, загляните на следующие ресурсы:

- » Страница Digia Qt — <http://qt.digia.com>
- » Домашняя страница проекта Qt — <http://qt-project.org>
- » Блог Qt — <http://blog.qt.digia.com>
- » Загрузка Qt — <http://qt-project.org/downloads>
- » Сообщество Qt — <http://planet.qt-project.org>



Что за штука... Qt5?

Марко Фиоретти выясняет, что скрывается за блестящей наружностью нового релиза.

В Просветите нас, что же такое Qt5? Вот бы это была какая-нибудь забавная игра слов, а не очередная нудная аббревиатура!

О Qt5 — последняя версия одноименного фреймворка, позволяющего разработчикам легко и быстро создавать приложения и графические интерфейсы для множества платформ. Название происходит от Quasar Toolkit, что по-своему забавно, на мой взгляд, ведь когда вы вслух произносите Qt, это звучит как 'cutie [англ. «милашка»]'. Хотя, согласно официальному сайту, должно быть 'cute [англ. «круто»]'.
В Как-то не особо забавно. Вы упомянули, что его можно использовать на многих платформах — насколько он гибкий?

О Говоря о кроссплатформенности Qt, его разработчики, Digia и команда Qt, определенно не шутят. Помимо Linux, вы можете создавать приложения для Windows, Mac, Android, iOS, BlackBerry, QNX, Sailfish, Solaris и Integrity.

В Вот это да. Немало. И для чего они могут использоваться?

О В основе Qt Project лежит идея создавать красивые, надежные и удобные приложения для любых поддерживаемых платформ. Разработчикам предоставляется доступ к библиотекам Qt на C++, что позволяет создавать сложные приложения с приятным и простым в использовании интерфейсом. Благодаря поддержке технологии ускорения графики, системы частиц и построения

теней OpenGL/OpenGLES, а также Qt Multimedia и инструментам Qt Graphical Effects, разработчики могут производить на свет действительно красивые приложения.

В Тогда понятнее, почему «милашка». Так что же нового в 5-й версии?

О Вышеперечисленные графические инструменты и есть главное новшество Qt5. Scene Graph [Граф сцены] на базе OpenGL здесь используется для расширения возможностей Qt Quick, создавая великолепную анимацию, графические эффекты и систему частиц даже без мощной аппаратной части. Это особенно актуально для написания элегантных приложений для мобильных устройств.

В Звучит здорово, но есть ли уже какие-то готовые примеры графического мастерства Qt5, на которые можно взглянуть?

О Конечно, есть. Зайдите на <http://qt-project.org/videos/watch/livecoding-video-effects-with-qt5>, чтобы увидеть пример создания видео-эффектов с качеством 720 p при помощи Qt5 в режиме live и понаблюдать, насколько быстро и легко можно их добавлять и настраивать. В Qt Blog (<http://blog.qt.digia.com>) есть прекрасная демонстрация применения эффектов построения теней для видео, с использованием Qt Quick 2, являющегося ядром Qt5. Найти ее можно по адресу <http://blog.qt.digia.com/blog/2012/02/29/pimp-my-video-shader-effects-and-multimedia>.

В Стало быть, Qt Quick 2 — важная составляющая Qt5? Какие же новоиспеченные функции это нам обещает?

О Qt Quick — основная библиотека для написания приложений QML, с особым упором на создание пользовательских интерфейсов. Последняя версия Qt Quick, 2.0, отказывается от классов Qdeclarative из Qt Quick 1 в пользу нового набора классов C++. Помимо расширенных возможностей анимации и графических инструментов, о которых мы сказали, добавлены встроенный Scene Graph и Canvas API, с помощью которых великолепные интерфейсы пользователя удаются разработчикам максимально легко, независимо от того, для какой платформы они предназначены. Все важнейшие оптимизации JavaScript для QML также учтены. Загляните на <http://qt-project.org/videos/watch/whats-new-in-qtquick-2.0>, чтобы посмотреть видео о некоторых новых возможностях Qt Quick 2.

В Ну вот, аббревиатур опять немеряно — и никаких мало-мальски забавных названий. Что это за QML?

О QML расшифровывается либо как Qt Meta Language, либо Qt Modeling Language, смотря кого спросить. Помимо дискуссий о том, от чего это сокращается — а не как читается, на сей раз — создатели QML разработали этот основанный на JavaScript язык для написания приложений, где во главе угла стоит интерфейс пользователя.

Поскольку этим ключевым компонентом Qt Quick ранее занималась Nokia, упор, безусловно, делается на мобильные приложения, в особенности для устройств с сенсорным экраном, где удобство (а также скорость и размер) интерфейса являются приоритетными.

Документы QML состоят из элементов QML и представляют собой мощные инструменты для создания и контролирования как графических, так и поведенческих свойств объектов. Связанные между собой, эти элементы способны создавать компоненты различной сложности — от простых кнопок до полноценных программ с доступом в Интернет. Кроме того, в целях еще большего

увеличения гибкости, QML (и, как следствие, Qt Quick и Qt5) могут быть дополнены стандартным JavaScript и объединены с компонентами C++.

В Ого, похоже, гибкость здесь в основе всего. Однако меня здесь смутило слово «Nokia». Какое отношение этот никем не любимый, почти тут непричастный производитель мобильных имеет к Qt5?

О К счастью, теперь почти никакого. В 2008 Nokia приобрела Trolltech, первоначального автора Qt, отдав за это около €104 миллиона и переименовав компанию в Qt Development Frameworks. Целью было возродить приложения Nokia на многострадальной мобильной ОС Symbian,

дающих разработчикам инструментарий для написания по-настоящему крутых приложений под любую платформу.

В Какие же еще модули есть в Qt5 и что они умеют?

О В Qt5 два основных типа модулей: Qt Essentials и Qt Add-ons. Первые несут характер основных и используются в большинстве приложений Qt. Это модуль Qt Core, от которого зависят все остальные, Qt GUI, включающий классы для компонентов графических интерфейсов, Qt Multimedia, отвечающий за функции аудио, видео, радио и фото, а также уже упомянутый Qt WebKit и многие другие.

«Создавать красивые, надежные и удобные приложения для любых платформ.»

с помощью удобного инструментария Qt. Но всего несколько лет спустя Nokia, наконец, швыряет истерзанную Symbian в могилку рядом с давно почившими WebOS и PalmOS и продает Qt Development Frameworks дружественной финской же компании Digia, где ею занимаются по сей день (см. подробнее: «Qt изнутри», на стр. 50).

В Ну да. А Nokia потом, я полагаю, продолжала пользоваться новой, успешной и готовой покорить весь мир мобильной ОС?

О Кхе, не совсем. Теперь они создают телефоны исключительно для Windows Phone 8. И тогда как Android и iOS поддерживаются Qt5, Windows Phone обошли стороной. Некоторые довольно крупные производители пытались найти способы портировать свои приложения Qt на Windows Phone, продвигаясь к преодолению одного из самых радикальных OpenGL-фобных (так сказать) барьеров среди мобильных ОС. Однако пока Microsoft не откажется от своей позиции «DirectX или пошли вон», перенести приложения Qt на Windows Phone будет по-прежнему проблематично, а значит, операционная система, которой без крутых приложений не выжить, останется без приложений, популярных на других платформах.

В То есть Qt5 нужен только для мобильных приложений?

О Вовсе нет. Для сетевых прекрасно подойдет Qt WebKit — движок для работы с web-контентом. В его основе лежит открытый проект WebKit, позволяющий разработчикам использовать Qt5 для поддержки самых разнообразных сетевых технологий и стандартов. Qt WebKit добавляет к нему HTML5 и весь его арсенал, включая фильтры CSS, WebGL, поддержку Canvas и HTML видео. Qt WebKit — один из множества модулей Qt5,

модули же Qt Add-ons служат для более конкретных целей. Это может быть обеспечение совместимости языков на разных платформах, как, например, в модуле Active Qt, предназначенном для платформ Windows и включающем классы для приложений, использующих ActiveX и COM. Кстати, модуль Qt D-Bus отвечает за межпроцессное взаимодействие посредством протокола D-Bus на платформах Unix. В составе Qt SVG — классы для отображения содержимого файлов SVG, а Qt Script и Qt Script Tools включают классы для создания сценариев в приложениях Qt.

Кроме того, среди модулей Add-on есть также Qt Declarative, обеспечивающий совместимость с Qt4 и Qt Print Support, который, согласно названию, позволяет разработчикам вставлять в приложения функцию печати. За красоту отвечают Qt Graphical Effects, Qt Image Formats и Qt OpenGL.

В И что за приложения написаны с помощью Qt и Qt5? Есть ли среди них те, о которых нам уже доводилось слышать?

О В числе наиболее известных приложений, созданных с использованием языка Qt или его элементов — Google Earth, VLC Media Player, Guitar Pro, Scribus, Last.fm, Launchy и Spotify. Элементы Qt есть даже в некоторых играх, например, в City of Heroes.

Открыв в браузере <http://qt-apps.org>, вы получите список приложений Qt, находящихся в стадии разработки. Только не смотрите комментарии внизу страницы — к сожалению, там полно спама. Естественно, то, что все это написано на Qt, не значит, что везде использовался Qt5. Однако по мере того, как все больше разработчиков обновляются до Qt5, вскоре мы увидим и улучшенные — или совсем новые — приложения, реализующие все его преимущества. **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.

Тест Тьюринга

Электрического монаха, который верил вместо вас, избавляя вас от упражнений в религиозности, для экономии усилий изобрел Дуглас Адамс [Douglas Adams].

Последовав его примеру, я изобрел ЭлектроТвиттер — программу, которая пишет за вас твиты. Я сделал это шутки ради, но обнаружил, что такая же (или почти такая) программа уже есть — AutoTweeter (www.autotweeter.in). Достаточно записать свои твиты в таблицу, и программа будет их отправлять. И уже не трудиться — всего за несколько долларов можно купить несколько тысяч готовых твитов.

Я представлял себе более сложную программу, с анализом предыдущих твитов, переписки и истории браузера, и чтобы она придумывала твиты сама. И я задумался: трудно ли пройти тест Тьюринга на твиты? (Тест Тьюринга — это проверка возможности компьютера обнаруживать интеллектуальное поведение, неотличимое от человеческого. Человек и компьютер обмениваются текстовыми сообщениями. Другой человек наблюдает за этим. Тест Тьюринга считается пройденным, если наблюдатель не может уверенно отличить, кто есть кто).

А сейчас я пишу Электрочитателя твитов, маленькую программу, которая читает ваши твиты и отправляет их в `/dev/null`. Если эти две программы заработают, их активность начнет бурно расти и выйдет из-под контроля, Интернет с грохотом взорвется, и мы наконец заживем нормальной, настоящей жизнью.

chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

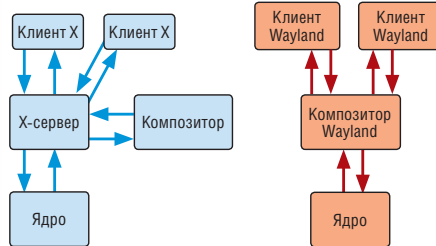


Композитор — что это?

Аккуратно обходя спор *Mir* и *Wayland*, Доктор отвечает на этот животрепещущий вопрос.

Спор *Mir* и *Wayland* привлек внимание масс-медиа недавно, когда Intel сообщила, что не примет заплатку к видеодрайверу Xorg для поддержки *Mir*. Это было пощечиной Sapnical. Я упоминаю об этом потому, что не хочу ввязываться в спор. *Mir* и *Wayland* — композитные графические серверы, и я осознал, что не до конца понимаю, что делает композитор. И решил, что пора это выяснить...

В традиционной оконной среде X каждая графическая программа (клиент X) отвечает за прорисовку своего окна и за перерисовку своего окна (или его частей) при перемещении или закрытии других окон, которые его перекрывают. При использовании же композитора каждое приложение хранит свое окно в буфере, а за прорисовку окон всех приложений на экране несет ответственность композитор.



В современных системах композитор работает вместе с X, но *Wayland* и *Mir* могут обходиться без X-сервера.

Композитор способен сделать окно полупрозрачным, изменить его размер, повернуть его, спроецировать на грани вращающегося куба, деформировать как мыльную пленку, положить под цифровое увеличительное стекло или заставить вылетать из иконки или с панели задач, как джинн из лампы. Во всех этих случаях приложение не знает, что творится с его окном.

Композиторы существуют уже довольно давно; самые популярные — *compiz*, *Xfwm* и *Enlightenment*. В Mac OS X есть *Quartz*. Но сейчас композитор работает поверх X, механизма прорисовки растрового экрана, целую вечность применяемого в Linux (а до этого в UNIX). Для работы с трехмерным аппаратным ускорением современных видеокарт протокол X-сервера был дополнен. Но *Wayland* и *Mir* могут совсем обойтись без X — для классической системы это будет шоком, но тем, кому нужна легкая система для планшетов и телефонов, будет удобнее.

На сайте *Wayland* есть хорошее обсуждение ограничений отдельных архитектур «X Server + композитор» и преимуществ их объединения (<http://wayland.freedesktop.org/architecture.html>).

Композитинг в кино

В кино понятие «композитинг» относится к объединению отдельных элементов в единую сцену, обычно — действий на переднем плане с другим фоном. Другие (оптические) технологии включают рипроекцию, многократное экспонирование

с использованием каше́ и «съемку со стеклянным фоном». К более свежим технологиям относится «хромакей», когда сплошной фон сцены заменяют изображением. Эта технология часто применяется для создания «виртуальных» телестудий.

Все на свете — файл

Надев резиновые сапоги и вооружившись факелом, бесстрашный Доктор отправляется исследовать файловую систему, чтобы узнать, кто в ней живет.

Запустите команду `ls -l` в любом каталоге, и большинство файлов, которые вы увидите, будут... просто файлами. Но в файловой системе живут и другие существа, и в этом месяце я отправляюсь в путь, чтобы их найти.

Обычные файлы

Старые добрые файлы — вид доминирующий, и вы, наверное, неплохо представляете себе, что такое файл. Но даже здесь кое-что может вас удивить. Например, ни один из файлов в каталогах `/proc` и `/sys` не существует — в том смысле, что их нет на диске. Эти файлы — целиком и полностью плод воображения ядра: это представление внутренних данных ядра в виде файлов. Чтобы понять это, нужно расширить ваше представление о том, что такое файл.

С точки зрения программиста, программа обращается к файлу с помощью четырех системных вызовов: `open()`, `read()`, `write()` и `close()`. Если ядро в ответ на вызов `read()` возвращает данные, то для программы пространства пользователя это и вправду файл, и его можно просмотреть обычными программами вроде `cat` и `less`.

Попробуйте скомандовать

```
cat /proc/cpuinfo
```

— и произойдет именно это. Но вы можете сказать, что происходит нечто странное, потому что если попробовать выполнить

```
ls -l /proc/cpuinfo
```

она покажет нулевую длину (явная ложь), и время изменения файла всегда будет совпадать с текущим временем.

Каталоги

Вы, вероятно, думаете, что каталоги — это контейнеры для других файлов, и это так, хотя и не совсем. Физически файлы не находятся внутри каталогов, как кофе у меня в чашке. Каталоги просто содержат ссылки, указывающие на файлы. Именно ссылки дают файлам имена. Поэтому, например, если переместить файл `foo` на уровень выше в иерархии каталогов —

```
$ mv foo ..
```

то данные на самом деле не будут перемещаться между каталогами: переместится только ссылка (с одним исключением: если ссылка перемещается на другой раздел, то будут перемещены и данные). Аналогично, если создать еще одну ссылку на `foo` командой

```
$ ln foo bar
```

то второй копии файла не создается, файлу просто выдается второе имя. Если изменить файл `foo` и затем посмотреть на файл `bar`, то вы увидите изменения. Файл всего один. Имя моего младшего сына — Мэтью, но друзья зовут его Мэт. Это не два разных человека, а один с двумя именами. Если я дам Мэту чашку кофе, Мэтью тоже взбодрится.

Когда я рассказываю о файловых системах в классе, все идет хорошо, пока мы не доходим до команды `ln`. Потом лбы начинают морщиться, и появляются вопросы. Обычно студенты спрашивают: «Зачем файлу несколько имен?» Ну, вот пример, опять же из учебного класса. В курсе по скриптам оболочки есть пример с файла с данными `shopping.txt`, используемый в нескольких частях. Примеры для каждой части находятся в собственном каталоге: `chap1`, `chap2` и т. д. Чтобы не копировать файл `shopping` в каждый каталог, у нас всего одна копия, а в каждом каталоге есть

Подсчет файлов

Чтобы найти эти загадочные существа в файловой системе, вам вообще-то не нужны ни факел, ни резиновые сапоги. Их легко обнаружить командой `find`. Например, для поиска сокетов выполните команду

```
# find / -type s
```

На основе этой команды я написал небольшой скрипт, который определяет

«частоту появления» файлов каждого типа и выводит их в виде таблицы:

```
for t in f d l b c p s
do
echo -n $t :
find / -type $t 2>/dev/null | wc -l
done
```

Пробуя скрипт у себя, запускайте его как `root`, иначе не зайти во все каталоги.

ссылка на него. Еще один пример: в моей Ubuntu имена `/usr/bin/pstruct` и `/usr/bin/c2ph` — это ссылки на одну и ту же программу. Программа (которая делает что-то крайне маловразумительное) ведет себя немного иначе в зависимости от того, по какому имени ее вызывают.

Впрочем, надо признать, что файлов с несколькими ссылками крайне мало, и я бы не стал преувеличивать достоинства такой схемы.

Кроме того, есть пара вещей, которые нельзя сделать с жесткими ссылками. Во-первых, нельзя создать второе имя для каталога:

```
$ ln /home myhome
ln: '/home': hard link not allowed for directory
```

Такое ограничение не случайно. По большому счету, именно это правило гарантирует, что структура файловой системы на самом деле будет деревом, а не произвольным образом переплетенной сетью каталогов.

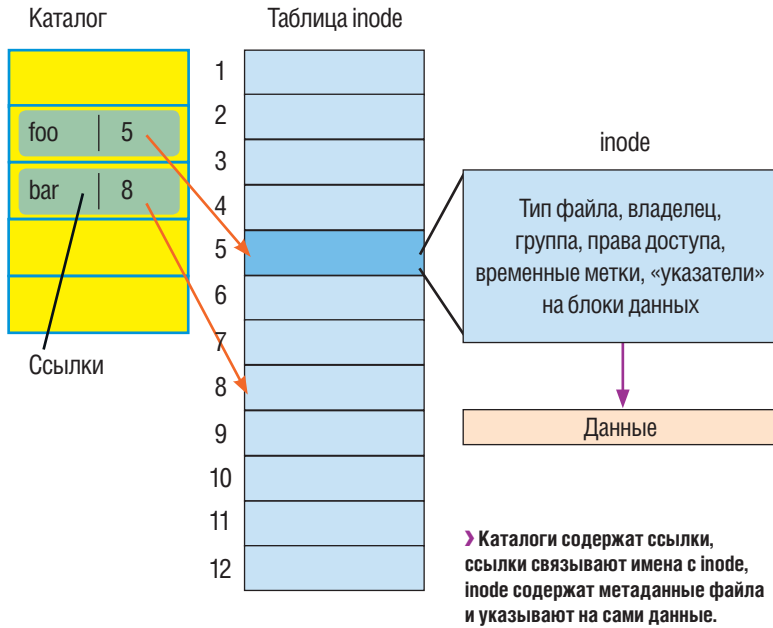
Второе, чего нельзя делать с жесткими ссылками — ссылаться с раздела на раздел. То есть, нельзя поместить в каталог на одном разделе ссылку на файл на другом разделе:

```
$ ln /etc/hosts myhosts
ln: creating hard link `myhosts' => `/etc/hosts': Invalid crossdevice link
```

»

Тип	Описание	Частота появления (Ubuntu)	Частота появления (CentOS)
—	Обычный файл (жесткая ссылка)	837239	164232
d	Каталог	154556	20809
l	Символическая ссылка	126858	21184
b	Блочное устройство	31	31
c	Символьное устройство	212	149
p	Именованный канал	0	4
s	Доменный сокет UNIX	26	101

► Семь типов файлов в файловой системе Linux и частота их появления в двух взятых для примера дистрибутивах Linux.



Здесь я пытаюсь поместить в свой домашний каталог (который находится на другом разделе) ссылку на файл в корневом разделе. Так сделать нельзя.

Немного подумав, вы поймете смысл ограничения. Ссылка преобразует имя в номер inode, но нельзя сказать «эта ссылка foo ссылается на inode номер 14328 на другом разделе вон там».

Что приводит нас к главе...

Символические ссылки

Символические ссылки (также часто называемые «symlink» или мягкими ссылками) сильно отличаются от жестких. Символическая ссылка — это самостоятельный маленький файл, содержимое которого — имя другого файла, обычно называемого «целью» символической ссылки.

Символические ссылки немного похожи на ярлыки в Windows. Благодаря своему устройству они позволяют нарушать правила (или, если угодно, обходить ограничения), установленные жесткими ссылками. Обе следующие команды сработают:

```
$ ln -s /etc/myetc
$ ln -s /etc/hosts myhosts
```

А просмотреть символические ссылки можно так:

```
$ ls -l my*
lrwxrwxrwx. 1 chris chris 4 Sep 8 19:08 myetc -> /etc
lrwxrwxrwx. 1 chris chris 10 Sep 8 19:08 myhosts -> /etc/hosts
```

Обратите внимание на l в начале, а также на права доступа **rw**. Насколько я знаю, эти права доступа ничего не значат — важны права доступа целевого файла.

Все это мило и интересно, но символические ссылки порождают некоторые вопросы. Если указать символическую ссылку в аргументе команды, она должна работать со ссылкой или с файлом, на который она ссылается? Ответ — «по-разному», и большинство команд работают «по уму». Например, **rm** удалит символическую ссылку, а не целевой файл, а **chmod** работает

с целевым файлом, а не с символической ссылкой. У некоторых команд есть поведение по умолчанию, которое можно изменить через параметры. Например, по умолчанию ни **ls**, ни **tar** не переходят по символическим ссылкам, но у них есть параметры (**-H** и **-h** соответственно), которые просят их сделать это.

Символические ссылки могут учинять и другие сюрпризы. Например, если я попытаюсь перейти в каталог, указав в качестве каталога ранее созданную ссылку **myetc** —

```
$ cd myetc
$ pwd
/home/chris/myetc
```

но на самом деле мы в **/etc**. Это сбивает с толку. Если удалить файл, на который ссылается ссылка, тоже возникнет странная ситуация. Ссылка станет недействительна. На самом деле, **ln -s** перед созданием ссылки даже не проверяет, что целевой файл существует. Эта команда «работает»:

```
$ ln -s xxx yyy
хотя файла xxx на свете нет!
```

Наконец, можно вызвать хаос, создав в каталоге символическую ссылку, которая указывает на каталог верхнего уровня в дереве: в дереве файловой системы получится петля, которая может как минимум озадачить.

Если у вас складывается впечатление, что я неравнодушен к символическим ссылкам, это так! Применяя их с умом и расчетом, можно делать удобные вещи, которые не сделать никак иначе. Но если отнестись к ним безответственно, головной боли не избежать.

Взгляните на таблицу с реальным количеством файлов каждого типа в двух системах: в моей повседневной Ubuntu и в свежеставленной CentOS. Различие между этими цифрами, вероятно, отражает количество пользовательских файлов и пакетов в этих системах, а не фундаментальные отличия двух дистрибутивов. Впрочем, ясно одно: обычные файлы, каталоги и символические ссылки — это 99,9% (или более) всех файлов.

Должен признать, что я был сильно удивлен количеством найденных символических ссылок. Лично я считаю их таким источником путаницы, что правительству стоит ввести небольшой налог на их применение. Немного — может, всего по пенни за ссылку — но достаточно, чтобы вы спросили себя: «А нужны ли мне они на самом деле?»

Блочные и символьные устройства

Пора перейти к четырем оставшимся, более редким типам файлов. О блочных и символьных устройствах я расскажу вместе. Эти ссылки дают имена устройствам, и почти все они находятся в каталоге **/dev** (сокращение от «devices» — устройства). Рассмотрим пару примеров:

```
# cd /dev
# ls -l sda1 console
crw-----. 1 root root 5, 1 Sep 8 18:54 console
brw-rw----. 1 root disk 8, 1 Sep 8 18:54 sda1
```

Обратите внимание на c и b в начале, а также на две пары чисел там, где обычно находится объем файлов. Эти числа называются старшим и младшим номерами устройства, и по ним

Ссылки и каталоги

Если вывести подробный список содержимого каталога, часто можно увидеть много ссылок, например:

```
$ ls -ld /usr/lib
drwxr-xr-x 220 root root 49152 Sep 7 06:42 /usr/lib
```

В свете моих комментариев о невозможности создавать дополнительные ссылки на каталоги, откуда взялись эти 220 ссылок? Ну, одна из них — это ссылка на родительский каталог (в данном случае, **/usr**). Это ссылка, которая фактически

прикрепляет каталог к дереву файловой системы. Кроме того, в каждом каталоге имеется ссылка на сам каталог (**.**). Еще 218 ссылок — это ссылки на подкаталоги (**..**). Больше этим ссылкам взяться неоткуда.

определяется драйвер ядра, который будет использован для доступа к устройству. Вообще говоря, символьное устройство работает с последовательным побайтовым потоком данных, а блочное — с блоками, которые можно адресовать и использовать произвольным образом. Классические блочные устройства — дисковые разделы, но иногда поверх них используются другие программные блочные устройства, например, для логических томов, RAID-массива или зашифрованных разделов. Эти блочные устройства важны, потому что необходимы Linux для построения файловой системы.

Не все файлы в `/dev` представляют реальные устройства: там есть и программные. Самое известное, пожалуй, `/dev/null`, которое просто поглощает все, что в него записывается, и с ним удобно явным образом избавиться от потока данных. Иногда его называют «черной дырой». Также есть `/dev/zero`, которое формирует постоянный поток нулей, и (более сложные) `/dev/random` и `/dev/urandom`, формирующие поток случайных чисел. Все это — символьные устройства.

Именованные каналы

Именованные каналы встречаются в Linux так редко, что их следует отнести к видам, находящимся под угрозой исчезновения. На самом деле, в моей Ubuntu в дикой природе они вымерли, а в CentOS их всего четыре. С помощью именованных каналов один процесс может отправить поток данных другому. Им не требуется заранее это описывать, но необходимо согласовать имя канала.

Именованные каналы являются однонаправленными. У них есть вход, куда можно записывать данные, и выход, откуда можно их считывать. Вот небольшой эксперимент для иллюстрации. Откройте два терминала и расположите их рядом. В левом терминале создайте канал:

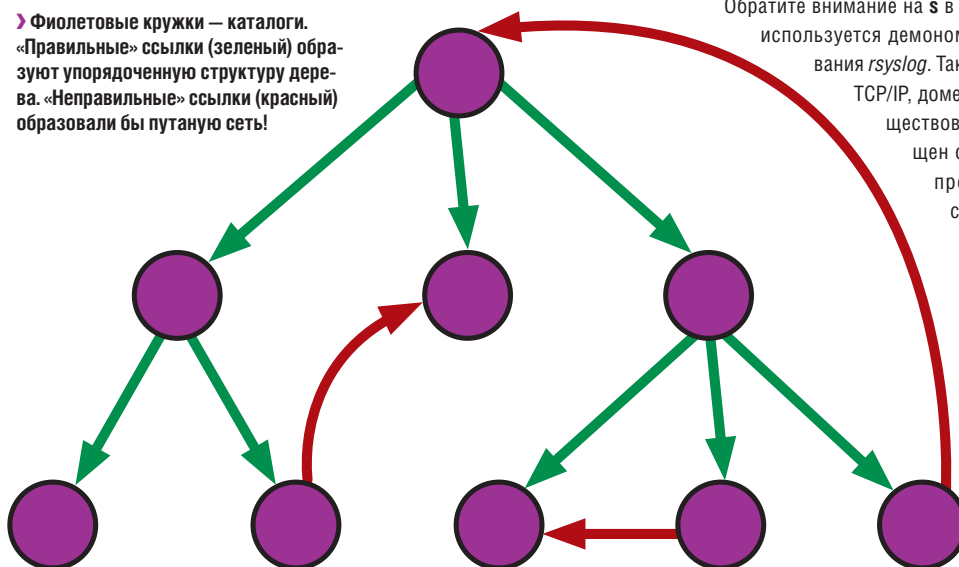
```
$ mkfifo /tmp/mypipe
$ ls -l /tmp/mypipe
prw-rw-r-- 1 chris chris 0 Sep 9 06:45 /tmp/mypipe
```

Обратите внимание на команду для создания канала — `mkfifo`. (FIFO — сокращение от “first in, first out” — «первый вошел, первый вышел»). Также отметьте начальную `p` в выводе `ls -l`.

В левом окне скомаундите

```
cat > /tmp/mypipe
Эта команда заблокирует командную строку, ожидая, пока вы наберете что-нибудь на клавиатуре. Теперь выполните следующую команду в правом терминале:
cat < /tmp/mypipe
```

► Фиолетовые кружки — каталогии. «Правильные» ссылки (зеленый) образуют упорядоченную структуру дерева. «Неправильные» ссылки (красный) образовали бы путаную сеть!



Все на свете — файл, кроме...

Исключение из правила «все на свете — файл», которое приходит на ум — сетевые интерфейсы, обычно называемые `eth0`, `eth1`, `wlan0` и т. д. Им не соответствуют файлы в файловой системе. Если вам придет в голову еще что-то с именем, но без файла, пожалуйста, сообщите мне на chris.linuxformat@gmail.com.

Она тоже заблокирует командную строку, ожидая появления данных в канале. Теперь начинайте набирать текст в левом окне. Текст будет записываться во вход канала первой командой `cat`, считываться из выхода канала второй командой `cat` и отображаться в правом окне. Это будет продолжаться до тех пор, пока вы не нажмете `Ctrl+D` в левом окне. В этот момент `cat` на входе канала завершится, и `cat` на выходе канала, обнаружив, что процесс на входе канала завершился, завершится тоже.

Также бывают анонимные каналы, но они не относятся к нашей теме, потому что им не соответствует файл в файловой системе (оттого они и называются анонимными). Чтобы два процесса могли взаимодействовать через анонимный канал, у них должен быть общий родитель, который создает канал и передает вход и выход канала детям. Именно это происходит, когда мы пользуемся каналом в командной строке.

Сокеты

Наконец, мы перешли к сокетам. Сокеты, живущие в файловой системе, иногда называют доменными сокетами UNIX, чтобы отличить их от доменных сокетов TCP/IP, посредством которых программы взаимодействуют друг с другом в сети TCP/IP. Доменные сокеты UNIX также предоставляют средства межпроцессного взаимодействия, но они имеют более строгие ограничения, так как клиент и сервер должны находиться на одном и том же компьютере. В отличие от именованных каналов, сокеты являются двунаправленными, однако с двух сторон сокет работает несимметрично.

На одном конце (сервер) соединение пассивно слушается, на другом (клиент) происходят активные действия (подключение и т. д.).

Продемонстрировать сокеты в командной строке не так просто, но если вы хотите увидеть сокет в дикой природе, попробуйте следующее:

```
$ ls -l /dev/log
srw-rw-rw-. 1 root root 0 Sep 8 18:55 /dev/log
```

Обратите внимание на `s` в начале строки. Этот сокет используется демоном системного журналирования `rsyslog`. Также учтите, что, как и сокет TCP/IP, доменный сокет UNIX будет существовать до тех пор, пока запущен создавший его серверный процесс. Если остановить сервис `rsyslog`, сокет исчезнет (если вздумаете это проверять, не забудьте потом перезапустить сервис).

Ну, и на сей раз произошло кое-что совершенно нетипичное: темы для рассказа и место на статью закончились одновременно! Продолжение следует.

Приход зрелости

С развитием цифровых технологий они все чаще используются для намеренной имитации старых (худших?) аналоговых устройств. Простой пример — на экране монитора или смартфона можно разместить красивые часы со стрелками. Тем, кто счел часы с цифрами передовой идеей, такое может показаться ретроградством, но на самом деле это лишь демонстрирует, какими умными стали наши едичники и нолики.

Когда-то я посетил лекцию Сэма Леффлера [Sam Leffler] на встрече Группы пользователей UNIX в Кембридже, на которой он показал коротенькую компьютерную анимацию под названием «Приключения Андре и Уолли Б». Ее создатели затем основали Pixar, а сама анимация по нынешним стандартам была довольно примитивной, но в ней впервые использовалось размытие изображения с помощью компьютера. Леффлер говорил о том, как сложно выделить достаточно циклов процессора, чтобы сгенерировать это размытие. Говоря о компьютерных мультфильмах, недавно мы с внучкой смотрели «Храбрую сердцем». Есть сцены, в которых фокус «сдвигается» для переключения внимания между персонажами на переднем и на заднем плане. В обычном кино движение фокуса — обычный способ эксплуатации ограниченной глубины резкости оптической фотографии, но в картинке, созданной компьютером, понятия глубины резкости нет, и для его имитации, естественно, требуется много циклов процессора. Опять же, нам приходится сводить новую технологию со своего обычного пути, чтобы имитировать старую.

А каков ваш любимый пример? Хотите узнать, как добавлять щелчки и помехи в свои аудиозаписи? Напишите мне на chris.linuxformat@gmail.com.

➤ Не то чтобы мило, но функционально. Здесь *Bleachbit* выводит список файлов в кэше Firefox перед их удалением.

Bleachbit

Доктор открывает удобную утилиту для прополки файловой системы и заметания следов.

Многие приложения, которыми мы пользуемся, хранят историю, кэши и логи — цифровые следы наших путешествий. Вряд ли кто-то из наших читателей делает нечто способное привлечь к нему внимание АНБ (извините, это фото с обнаженным Туксом не в счет). Однако на наших компьютерах просто много... э-э... мусора, который забивает диски, раздувает резервные копии и (вероятно) нарушает вашу частную жизнь.

Найти и удалить значительную часть этого мусора поможет утилита с интересным названием *Bleachbit*. Как написано на ее web-странице, *Bleachbit* «освобождает кэш, удаляет куки, очищает историю серфинга, уничтожает временные файлы, удаляет лог-файлы и ликвидирует мусор, про который вы даже можете и не знать».

Я установил ее в свою рабочую систему Ubuntu и был удивлен тем, как много мусора оставляют за собой программы. Кэш *Chrome*? 2336 файлов размером 2,09 ГБ. Индекс *recol*? 1,26 ГБ. Кэш миниатюр? 159,2 МБ. И так далее. Все это нарастает.

Пользовательский интерфейс *Bleachbit* представляет собой хорошо знакомый «проводник»: на левой панели показаны отдельные «места для

очистки», а на правой — их описание. Нажмите кнопку Preview [Предпросмотр], и вы увидите список файлов, которые можно удалить, и занимаемое ими место. Нажмите кнопку Clean [Очистить], чтобы перейти от слов к делу. Конечно, приложению нужно доверять, так же как вы доверяете человеку, который подготовил ваш парашют, перед прыжком с самолета.

Bleachbit также может уничтожить (перезаписать) отдельные файлы или каталоги (зайдите в меню File [Файл]), сделав их недоступными для любопытных глаз программ исследования компьютеров; впрочем, то же самое делает в командной строке команда `shred`.

«Приложения, которыми мы пользуемся, хранят историю.»

Думаю, 6 сентября, когда Брюс Шнайер [Bruce Schneier] рассказал о *Bleachbit* в своей колонке в *The Guardian* (<http://bit.ly/1fReVn2>), сайт *Bleachbit* пережил большой наплыв посетителей. Почему бы вам тоже не попробовать ее? Программа должна быть в репозиториях вашего дистрибутива; если нет, зайдите на <http://bleachbit.sourceforge.net>.

С командной строки

При запуске *bleachbit* без аргументов откроется графический интерфейс. Но все, что можно сделать в нем, можно сделать и в командной строке.

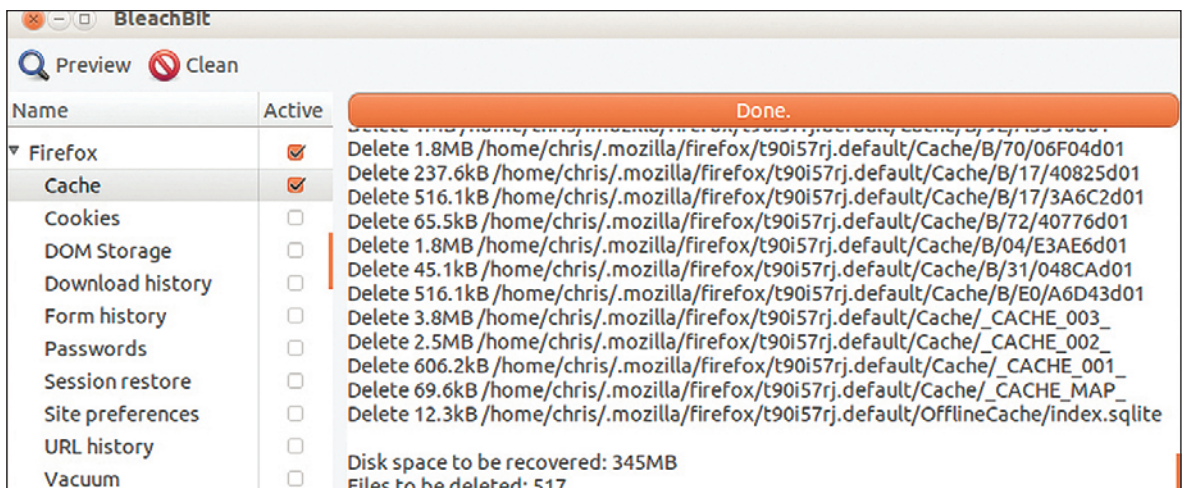
Например, для получения списка всех мест для очистки выполните команду

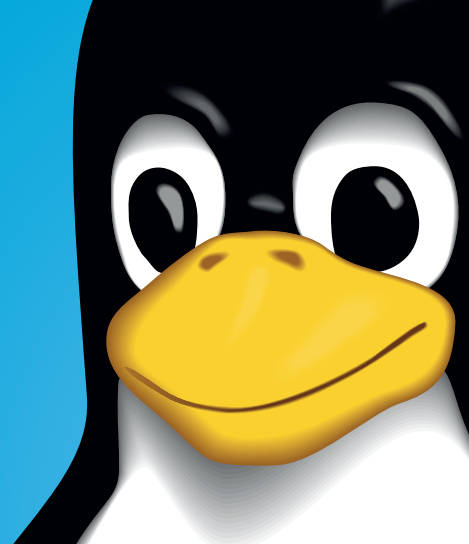
```
$ bleachbit --list
```

а для удаления всех файлов в кэше *Firefox* — команду:

```
$ bleachbit --delete firefox.cache
```

Утилиту можно прекрасно автоматизировать, добавив ее в скрипты или запуская как задание *Cron*.





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

Аварийная загрузка

Доктор впадает в легкий пессимизм и размышляет над тем, что делать, если система не загружается.

Итак, вы забыли свой пароль root? Фыркать не стоит: такое случается. Или система досталась от кого-то, кто забыл сообщить эту маленькую деталь? К счастью, не все потеряно! На этом уроке я покажу несколько способов восстановить систему.

Наряду с утратой пароля root, есть и другие ситуации, когда могут пригодиться эти методы. Например, ошибки в настройках *Grub* или в */etc/fstab* легко могут сделать систему незагружаемой.

Однопользовательская загрузка

Один из самых простых способов и определенно первый на очереди, чтобы попробовать — так называемая однопользовательская загрузка. Это проще сделать, чем описать в словах, но я рискну. Во-первых, прервите загрузку так, чтобы появилось меню *Grub*. (Обычно это окно появляется очень ненадолго, так что будьте бдительны!) Затем выберите пункт меню для той системы, которую хотите загрузить (ну или он всего один), и нажмите 'e', чтобы изменить команды. *Grub* покажет команды, соответствующие этому пункту меню. Выберите строку *kernel* и снова нажмите **e**, чтобы изменить ее; перейдите в конец этой строки и добавьте **s**. Этот параметр передается ядру и сообщает ему, что нужно загрузиться в однопользовательском режиме. Теперь нажмите Enter и затем **b**, чтобы загрузиться с этим дополнительным параметром.

В стандартной системе вроде RedHat при однопользовательской загрузке будут смонтированы файловые системы, но ни сеть, ни X-сервер не запустятся. Потом откроется оболочка от имени root! В этот момент можно задать новый пароль root командой **passwd** или исправить в настройках другую ошибку, которая была источником проблемы.

Некоторых администраторов беспокоит — и не зря — то, что с помощью этого простого и хорошо известного метода каждый может получить root-доступ. Это довольно легко поправить: в RedHat 6 и производных, которые используют *upstart*, найдите следующую строку в */etc/sysconfig/init*:

```
SINGLE=/sbin/sushell
и измените ее на
SINGLE=/sbin/sulogin
```

А в более старых системах, в которых еще используется */etc/inittab*, добавьте такую строку:



Утопающий пингвин? Позвольте усомниться. Настоящий Тукс утонуть не может.

```
~:~:wait:/sbin/sulogin
```

Хотя так мы повысим безопасность, мы лишим самих себя единственного (пока) способа восстановления утраченного пароля root. Есть и более экстремальный прием, который может сработать: прервите загрузку и отредактируйте параметры ядра точно так же, как прежде, но на сей раз добавьте аргумент **init=/bin/bash**. Этот параметр сообщает ядру, что вместо демона инициализации (*upstart*) нужно запустить оболочку. Linux моментально загрузится и перебросит вас в консоль с правами root. В этот момент система находится в очень сыром состоянии, и ее придется немного помассировать (в частности, вручную смонтировать файловые системы), чтобы перевести в то состояние, когда можно успешно выполнить команду **passwd**.

Аварийная загрузка

Если однопользовательская загрузка не решила проблемы, попробуйте аварийную. Идея в том, чтобы загрузиться в рабочую систему с минимумом содержимого с внешнего носителя, например, с CD или DVD (или USB-брелка), и хранить маленькую файловую систему в оперативной памяти. В этой системе можно смонтировать файловые системы на жесткий диск, чтобы посмотреть и (если повезет) исправить ошибки. Это гораздо более общий подход, и с его помощью можно исправить гораздо больше проблем.

Раньше для аварийной загрузки широко использовался Knoprix, так как он был одним из первых дистрибутивов с Live-загрузкой, но сейчас ее поддерживают множество дистрибутивов. В общем случае будет проще выполнять аварийную загрузку с тем же дистрибутивом, который вы пытаетесь спасти, но это не обязательное требование. Разумеется, есть множество возможных сочетаний спасателя и спасаемого, но в этом руководстве в качестве утопающей выступала нерабочая система CentOS, а в качестве спасателей — три разных системы: компакт-диск с настольной версией Ubuntu 12.04, специализированная среда восстановления *Rescatux* и аварийная загрузка с самой CentOS. Но я не буду их сравнивать и выставлять им оценки.

Чтобы выполнить аварийную загрузку, вам понадобится ISO-образ «спасателя». Запишите образ на компакт-диск и вставьте диск в привод (если вы спасаете физический компьютер) или

Проблемы SELinux

Если вы спасаете систему, где запущен SELinux (к ним относятся и RHEL 6, и ее производные), довольно легко создать неправильный контекст безопасности SELinux в таких файлах, как */etc/passwd* или */etc/shadow*, в результате чего при загрузке система не сможет получить к ним доступ — и войти в систему не сможет вообще никто! Если вы столкнулись с этой проблемой, загрузите SELinux в Разрешающем режиме [Permissive mode]. Для этого при загрузке добавьте параметр ядра **enforcing=0** (точно так же,

как добавляли параметр **s** для однопользовательской загрузки). Другой вариант — полностью отключить SELinux для этой загрузки, указав параметр ядра **selinux=0**.

Затем, когда вы попадете в командную строку, восстановите корректный контекст безопасности SELinux командой **restorecon**, например, таким образом:

```
# restorecon -v /etc/shadow
restorecon reset /etc/shadow
context system_u:object_r:file_t:s0-
>system_u:object_r:shadow_t:s0
```


подключите образ как виртуальный диск (если спасаете виртуальную машину). Теперь включите компьютер и измените настройки BIOS так, чтобы компьютер загружался с компакт-диска, а не с жесткого диска. В первом тесте я взял компакт-диск с настольной версией Ubuntu 12.04. Наберитесь терпения и наблюдайте, как величественно и неспешно он загружается. Наконец, появится окно, где можно выбрать, стоит ли пробовать Ubuntu или установить ее. Выберите Try Ubuntu [Попробовать Ubuntu]. В конце концов появится рабочий стол Ubuntu. Откройте терминал. (Если вы новичок в Unity, то сразу не поймете, как это сделать. Щелкните по иконке панели слева вверху и наберите “terminal” в строке поиска, затем щелкните на иконке терминала. Другой вариант — просто нажать Ctrl+Alt+T). После появления подсказки системы, следующий шаг — понять, где какой раздел на жестком диске; больше всего вас, вероятно, интересует корневой раздел.

Будучи старомодным типом, обычно я храню настройки системы в маленьком блокноте (чего и вам рекомендую!), но если вы делаете все вслепую, сейчас можно по крайней мере получить список разделов жесткого диска командой

```
$ sudo fdisk -l
или
$ cat /proc/partitions
```

Размеры разделов, которые сообщит команда, могут помочь догадаться о том, где какой раздел, но если нет — просто смонтируйте каждый раздел и посмотрите, что там!

Предположим, корневой раздел вы нашли. Дальше есть несколько вариантов подхода. Самый прямой — смонтировать раздел, например, так:

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

Теперь можно извлечь забытый пароль root, открыв файл shadow:

```
$ sudo vi /mnt/etc/shadow
```

Будьте осторожны — убедитесь, что вы редактируете файл shadow спасаемой, а не своей собственной системы. Найдите строку для root и сотрите хэш пароля (между первым и вторым двоеточиями). Сохраните файл командой w! и перезагрузитесь:

```
$ sudo shutdown -r now
```

Когда CentOS снова загрузится, вы должны войти в систему от имени root (или переключиться на него командой su) без пароля. Не стоит и говорить, что после этого первым делом нужно задать новый пароль root.

Альтернативный подход — переключиться в корневую файловую систему компьютера, который вы спасаете, командой chroot; после этого вы сможете задать пароль напрямую. Поэтому для задания нового пароля нужно вернуться в Ubuntu и выполнить команду

```
$ sudo chroot /mnt
а затем
# passwd
```

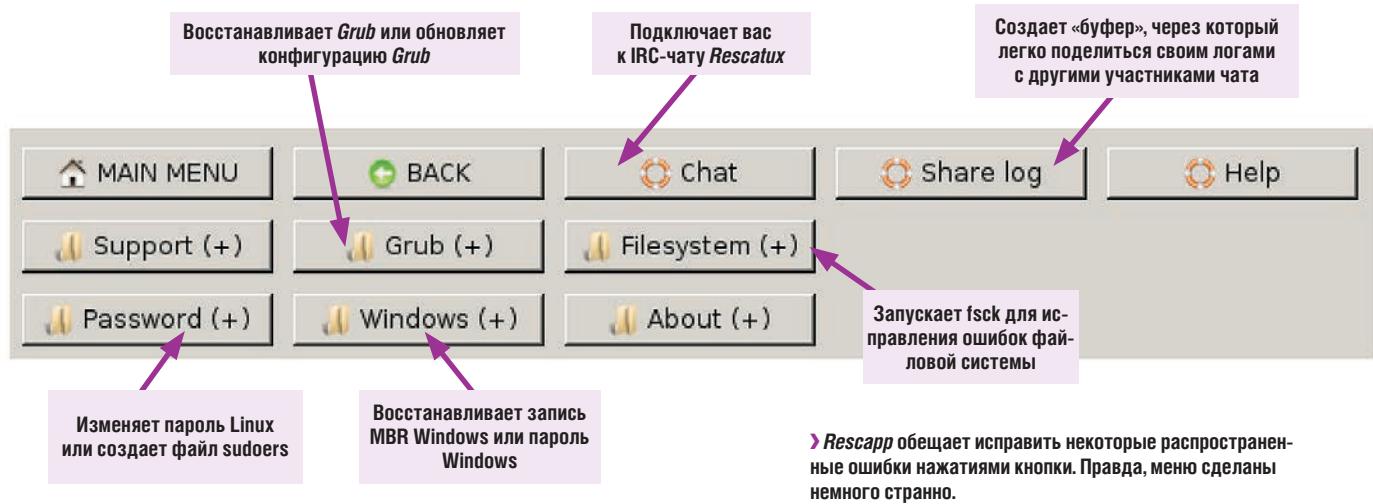
для установки нового пароля root.

Мир chroot — странный мир. Все еще работает ядро основной системы, но вы уже полностью полагаетесь на ядро спасаемой системы во всем остальном — в исполняемых файлах, таких как команда passwd, и любых разделяемых библиотеках, которые оно использует. Следовательно, в этом случае на компьютере, который вы спасаете, должна быть достаточно полная рабочая файловая система.

Тукс спешит на помощь

Хотелось бы подчеркнуть, что с помощью аварийной загрузки можно решить гораздо больший набор проблем, чем наши: забытый пароль — лишь верхушка айсберга проблем, которые можно ликвидировать. Так как можно смонтировать, осмотреть и изменить любую часть файловой системы захворавшего компьютера, то в принципе можно решить любую проблему, поддающуюся диагностике. На практике главную сложность обычно представляет именно установка диагноза.

Говоря об аварийной загрузке, нельзя не упомянуть Rescatux, удобный Live-дистрибутив на основе Debian и LXDE, предназначенный специально для решения проблем в неисправных системах (www.supergrubdisk.org/rescatux).



Спасение Ubuntu

Если у вас Ubuntu или одна из множества ее производных, вы вряд ли забудете пароль root, потому что тут его нет. И нельзя войти в систему под пользователем root, и нельзя переключиться на root командой su. Вместо этого в конфигурационном файле sudoers есть запись, которая

позволяет членам группы sudo (думаю, в более ранних версиях она называлась “admin”) выполнять любые команды с правами root, предварительно применив команду sudo.

Даже если в вашем дистрибутиве обычно все не так, большинство системных администраторов,

которые думают о безопасности, сами настраивают подобную схему.

Тем не менее, вы все равно можете лишиться себя возможности «переключиться на root» — для этого достаточно удалить или повредить файл /etc/sudoers!

Установка пароля Grub

Если вас лишает сна сама идея попадания в оболочку с root-доступом посредством прерывания загрузки *Grub*, подумайте о том, чтобы поставить пароль на сам загрузчик *Grub*.

При обычной загрузке (той, при которой используются настройки из конфигурационного файла *Grub*) вводить пароль не придется, но при прерывании загрузки

и изменении команд *Grub* (как я описал) — придется. Пароль *Grub* не обязан совпадать с паролем *root* — и, наверное, не должен бы.

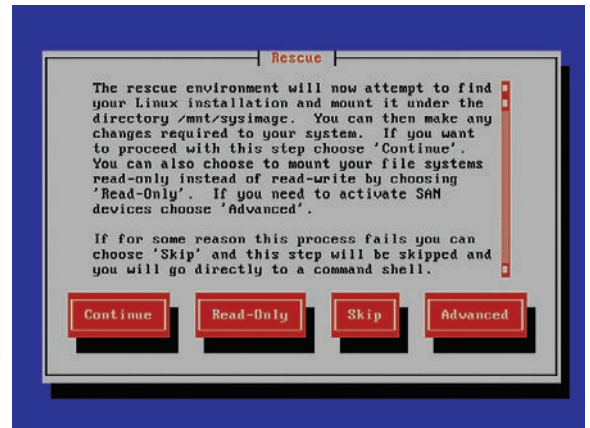
Помните о том, что сильнее затягивая гайки безопасности, вы можете усложнить восстановление системы. Если пароли *root* и *Grub* одинаковы и вы их забыли, вам все равно понадобится аварийная загрузка!

На данный момент *Rescatux* находится всего лишь в версии 0.30.2 и содержит довольно невыразительный набор утилит без всякого лоска. Но суть здесь не в самых свежих графических красотах. Основной предмет его гордости — утилита под названием *Rescapp*, с помощью которой можно исправить ошибки, не позволяющие загрузить систему или войти в нее, посредством мастеров. К таким ситуациям относятся забытые пароли, утраченные записи MBR и нерабочие файлы *sudoers*.

Итак, легко ли сбросить забытый пароль *root* с *Rescatux*? Довольно легко. Я насчитал девять щелчков мышью и несколько нажатий клавиш для ввода нового пароля. Загрузитесь с компакт-диска *Rescatux*, выберите *Password(+)* [Пароль (+)] из главного меню, затем *Change Gnu/Linux Password* [Изменить пароль Gnu/Linux] и *Change Gnu/Linux Password* [Изменить пароль Gnu/Linux]! Вас попросят выбрать раздел, где находится пароль, который нужно изменить, и пользователя, пароль которого нужно задать. Вводите пароль внимательно: он не отображается на экране и вводится всего один раз. Вот и все! Теперь просто перезагрузитесь.

Впрочем, конец этой истории с *Rescatux* был не совсем счастливым, потому что *Rescapp* ошиблась с контекстом SELinux файла */etc/shadow*, и это привело к тому, что после загрузки вообще никто не мог войти в систему. Это поправимо (см. врезку «Проблемы SELinux»), но усложняет то, что *Rescatux* пытается упростить.

Если не считать проблемы с SELinux, думаю, будет справедливым сказать, что восстановление пароля *root* с *Rescapp* проще, чем с обычной аварийной загрузкой... для этого не нужно знать ничего о файле *shadow*, и не требуется переключать файловую систему с *chroot*. Но, как и все мастера, он умеет выполнять лишь те фокусы, которым его научили. Ручное монтирование дисков, исследование и исправление ошибок на системных разделах — гораздо более общий подход. Я все еще твердо уверен, что секрет в исправлении какой-то ошибки в том, чтобы, во-первых, понять,



Аварийная загрузка CentOS автоматически обнаружит и смонтирует всю целевую файловую систему, и ее легко будет запустить в среде *chroot*.

как все должно было работать изначально, и во-вторых, действительно читать сообщения об ошибках.

А если у меня логические тома?

Чтобы сделать жизнь чуть интереснее, я решил повторить весь эксперимент в свежеставленной CentOS 6.4, корневой раздел которой находился на логическом томе, а не на сыром разделе. Это немного повышает ставки, так как основной системе теперь нужно определить логический раздел.

Live-дистрибутив Ubuntu добился немногого — пакета *lvm2* не было, и его было нужно установить из репозитория (помните, что это *live*-дистрибутив, а значит, установку придется делать каждый раз).

Хотя потом он смог найти физические тома, группу томов и логический том, файла устройства для логического тома не было, и я не смог его смонтировать. Но будем справедливы: это *live*-дистрибутив, который предназначен для краткого знакомства с настольной версией Ubuntu, а не для спасения системы.

Rescatux справился гораздо лучше: он распознал логический том, и я смог сбросить пароль с помощью *Rescapp* точно так же, как раньше. Я также смог вручную смонтировать логический том:

```
# mount /dev/vg_centos64/LogVol00 /mnt
```

и получить доступ к файловой системе CentOS. Опять же, с помощью волшебства *chroot* я могу сбросить пароль *root* в CentOS еще одним способом:

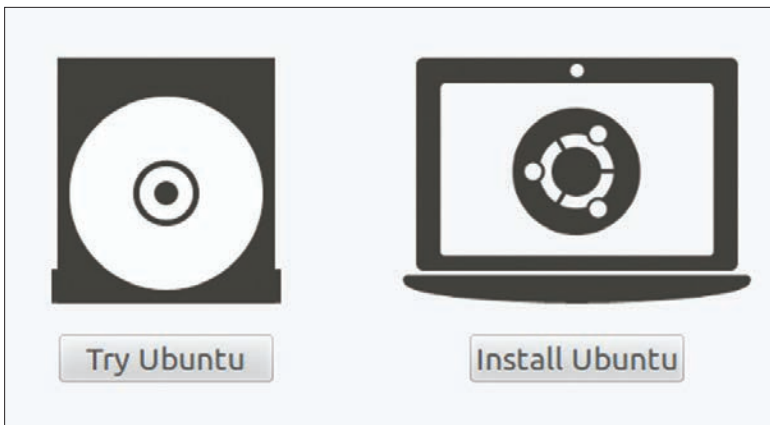
```
# chroot /mnt
```

```
# passwd
```

После этого я вправе установить новый пароль *root*.

Между установкой нового пароля утилитой *Rescapp* (которая выполняет команду *passwd* в системе *Rescatux*) и в среде *chroot* (которое я только что продемонстрировал) есть очень небольшое, но важное различие. Первая формирует свертку пароля MD5 (в полях с паролем */etc/shadow* начинается с *\$1\$...*); вторая — свертку SHA-512 (начинается с *\$6\$*). По этой причине *Rescapp* рекомендует еще раз сбросить пароль *root* после загрузки в спасаемую систему.

Наконец, я попробовал аварийную загрузку из самой CentOS. В ней нет мастера в стиле *Rescapp*, но она может автоматически найти существующую установку RedHat/CentOS, и CentOS автоматически смонтирует все части найденной файловой системы в соответствующие точки монтирования в */mnt/sysimage*. Так, например, она смонтировала мой раздел */boot* в */mnt/sysimage/boot*, а логический том с корневой файловой системой — в */mnt/sysimage*. Если переключиться на нее с помощью *chroot*, у вас появится доступ ко всей целевой файловой системе, и можно будет сбросить пароль без особого труда. LXF



Опция *Try Ubuntu* [Попробовать Ubuntu] установщика Ubuntu дает вам среду, которую можно использовать для восстановления в крайнем случае.

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Техническая литература

Будьте осторожны, читая книги о здоровье: опечатка может стоить вам жизни.
Сэмюэл Лэнгхорн Клеменс

Некоторые виды ПО фантастически сложны, причем сложно даже не их устройство, а интерфейс «рядового пользователя». Например, программный пакет *Code_Aster*, для структурного анализа посредством метода конечных элементов, имеет документацию в 17 000 страниц. Это более чем в 10 раз превышает число страниц в «Войне и мире» Л.Н. Толстого. Основной минус этого текстового богатства, что все на французском. Правда, есть плюс: само ПО является свободным. И если будет необходимо и появятся носители специализированных знаний, способных его использовать, то появятся и переводы. Хотя для возникновения практического интереса нужно развивать высокие технологии и производство.

Жан-Пьер Обри [Jean-Pierre Aubry] так и сделал, в смысле написал книгу “Beginning with Code_Aster” и даже выложил в открытый доступ ее электронную версию и исходники для тех, кто не может позволить себе заплатить €58. IMHO очень достойный поступок. Полку свободной технической литературы прибило.

Читатель, а ты планируешь написать что-либо полезное? В принципе, если да, можем поговорить об этом.

PS Немецкий суд признал недействительным патент на FAT. Честно говоря, неприятная была засада, т.к. FAT применялся чисто для совместимости. Стандарты и технологии, их реализующие, должны быть свободны!
e.m.baldin@inp.nsk.su

В этом месяце вы научитесь...



Упорядочивать фотки 66

Мэтт Хэнсон осваивает *Shotwell*, менеджер изображений номер один в Linux, чтобы щегольнуть своими «художественными» фотографиями.



Защищать серверы 68

Джонатан Робертс теперь работает всерьез, и он поделился с нами секретами управления серверами высокой доступности.



Добыывать память 72

На жестком диске легко накапливается всякий хлам, даже и без помощи операционной системы. **Нейл Ботвик** — поборник чистоты.



Мастерить рабочий стол 74

Призовем оконный менеджер, панель, избранные приложения и **Майка Сондерса** в помощь — и получим личный рабочий стол.



Резвиться с SSH 78

Желаете отправлять команды сразу на несколько компьютеров в удаленных точках земного шара (в чулане и под кроватью)? Позвольте **Нейлу Ботвику** научить вас применять SSH.



Разворачивать Drupal 80

По полной схеме задействуйте мощь системы управления контентом *Drupal* самым ленивым эффективным способом — от **Шашанка Эффективного (Шарма)**.



Мониторить сеть 84

Шашанк Шарма делится советами по *Zabbix*, удобному инструменту наблюдения за вашей сетью: это не просто броское наименование.



Работать с Git 88

Если вы — матерый линуксоид, то, несомненно, имели дело с Git, альтернативой BitKeeper от Линуса Торвальдса. **Джон Лэйн** дает пояснения.



Применять маски 92

Андрей Прахов научит вас рядиться в виртуальных монстриков — а может, и в отважных героев с мечами и бластерами; как кому нравится.

Shotwell. Рулим фотографиями

Мэтт Хэнсон гуляет по дружелюбному к пользователю фото-органайзеру, разбираясь, легко ли импортировать и редактировать отпускные фотографии.



Наш эксперт

Мэтт Хэнсон во-зился с операционными системами со времен Amiga Workbench 2.0. Покантовавшись в DOS и Windows, он обратил свое внимание на Linux.



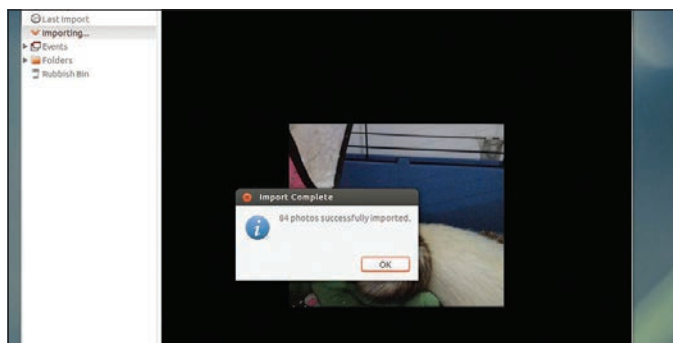
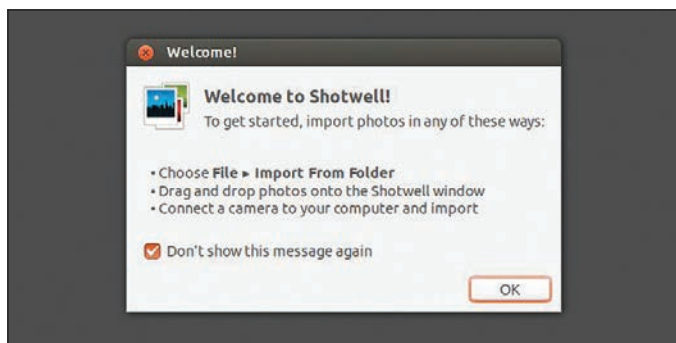
Появившись в 2009 году, *Shotwell* быстро завоевал репутацию отличной открытой альтернативы проприетарным программам управления фотографиями, таким, как *Windows Live Galley* и *iPhoto*. Хотя копирование фотографий с камеры или телефона на ПК — довольно простой процесс перетаскивания файлов, у *Shotwell* имеется симпатичный графический интерфейс пользователя, который позволяет быстро импортировать и организовывать свои фотографии. Вы можете группировать их по дате или по метаданным — например, по информации о географическом положении.

Помимо простого импорта ваших фотографий практически с любого устройства, которое вы подключите, можно быстро добавлять метаданные, присваивая файлам «тэги». Например, вы можете отметить фотографии из недавней поездки в солнечный Оксфорд тэгами «отпуск, Оксфорд». Это создаст два тэга — «отпуск», который будет относиться к группе любых отпускных фотографий, и «Оксфорд», что даст вам более узкий критерий поиска.

Shotwell также предлагает ряд базовых — хотя и всесторонних — инструментов редактирования фотографий, а также опции для быстрой публикации и выкладывания фото и видео на такие онлайн-сервисы, как Flickr и YouTube.

Короче говоря, *Shotwell* — отличное приложение для того, чтобы превратить нудный и длительный процесс в нечто быстрое и простое. LXF

Организируйте свои изображения в Shotwell

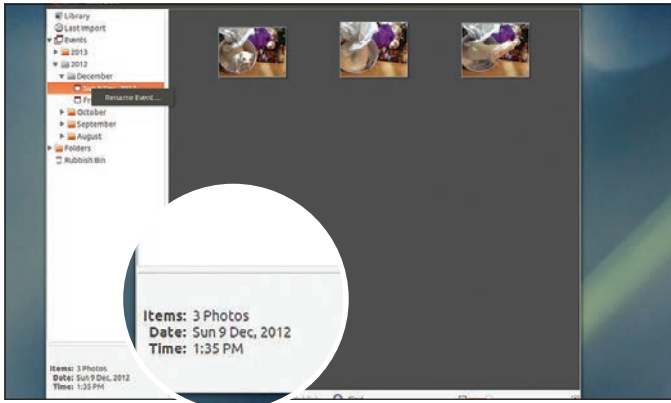


1 Импорт фото с ПК

Не так давно *Shotwell* заменил *F-Spot* в качестве стандартного инструмента работы с изображениями во многих дистрибутивах на базе Gnome, и весьма вероятно, что он у вас уже предустановлен. При самой первой загрузке вам предложат импортировать свои файлы из папки `home/user/Pictures`. Если ваши фотографии хранятся где-то еще, можете включить также и другие папки.

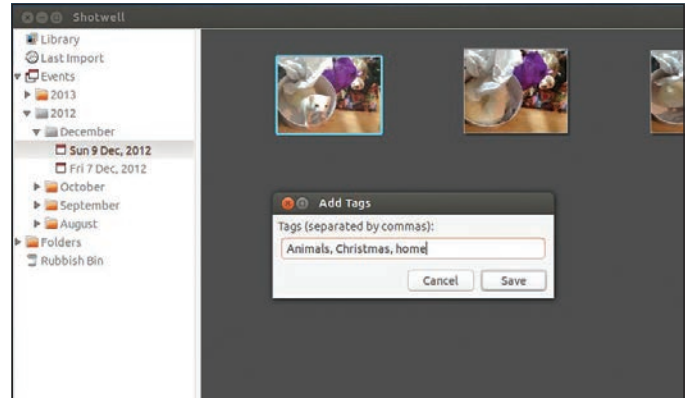
2 Импорт файлов с камеры или телефона

Некоторые цифровые камеры и почти все смартфоны идут с собственным раздутым редактором изображений и/или программой управления, которая настаивает на том, чтобы поселиться в вашем компьютере, став программой по умолчанию для любой задачи и для открытия разных типов файлов. К счастью, *Shotwell* может взять изображение и импортировать его прямо с вашего любимого устройства.



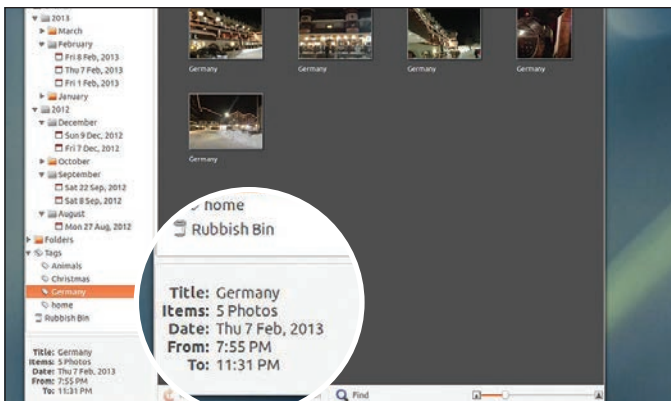
3 События Shotwell

Когда фото импортированы в *Shotwell*, они организуются по Событиям [Events], которыми по умолчанию являются дата и время, когда было сделано фото (или когда оно было импортировано, если информация по дате и времени отсутствует). Если щелкнуть правой кнопкой по событию, можно дать ему более запоминающееся имя.



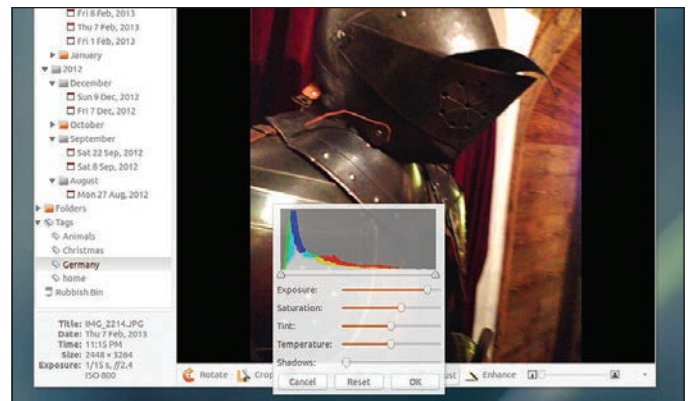
4 Добавление метаданных

Один из простейших способов добавления информации к вашим фотографиям для того, чтобы было проще их организовывать, это щелкнуть правой кнопкой по фото, выбрать *Add Tags* и затем через запятую ввести тэги. Чем больше тэгов вы добавите, тем проще вам будет находить фотографию впоследствии. Вы можете приписать тэг к нескольким фото, выделив их перед щелчком правой кнопкой.



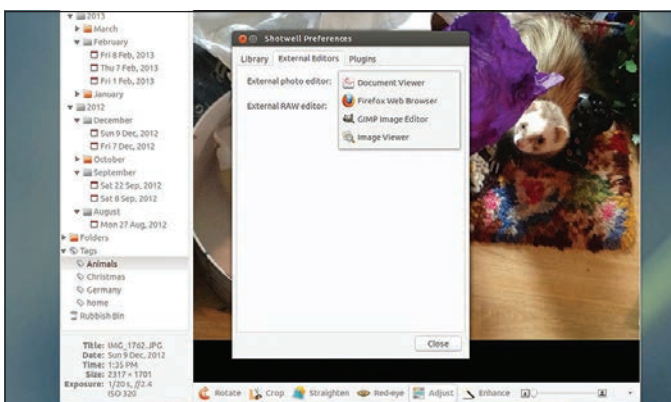
5 Поиск по метаданным

Вдоль левой стороны окна вы увидите разные события и даты, когда они произошли. Желая изменить данные о том, когда было сделано некое фото, щелкните по фото правой кнопкой и выберите *Adjust time and date* [Уточнить время и дату]. Все добавленные вами тэги появятся под датами. При щелчке по тэгу отобразятся фото, помеченные им.



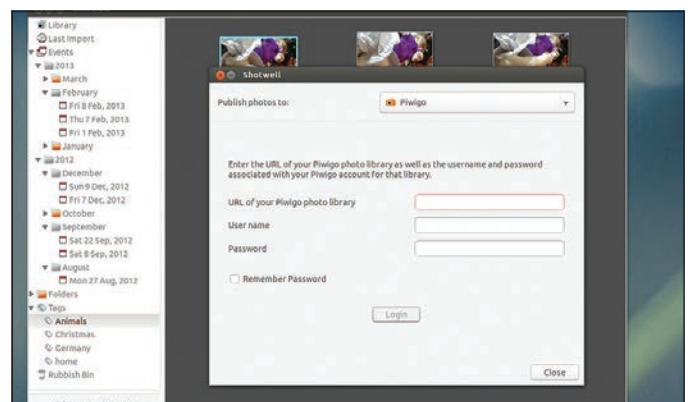
6 Редактирование фото

Дважды щелкните по фото, и вы сможете быстро отредактировать его с помощью ряда удобных инструментов. Инструменты *Rotation* [Поворот] и *Cropping* [Кадрирование] удобны для исправления плохо скомпонованных снимков, а инструменты *red eye reduction* [устранение эффекта красных глаз] и *выдержки*, *saturation* [насыщенности], *tint* [оттенка] и *temperature* [температуры] дают вам больше возможностей для улучшения качества ваших фотографий. Есть также кнопка *one-click fix* [починка одним щелчком], которая автоматически поправит ваши изображения.



7 Внешнее редактирование

Инструменты редактирования в *Shotwell* идеально подходят для быстрой правки, но если вы занимаетесь фотографией серьезно, вам понадобятся более мощные инструменты. Щелкните по *Edit* [Править] в панели инструментов, затем выберите *Preferences* [Параметры] и щелкните по *External Editors* [Внешние редакторы], чтобы выбрать другие программы, которые вы предпочитаете использовать, такие как *GIMP*, для редактирования своих фото и RAW-изображений.



8 Экспорт и публикация

Чтобы опубликовать или поделиться фото онлайн, выберите фото и щелкните по кнопке *Publish* [Опубликовать]. Список онлайн-сервисов, на которые можно выложить фото, находится в верхнем правом углу окна. Выбрав нужный сервис, щелкните по кнопке *login*, чтобы зайти на него. В зависимости от выбранного сервиса, вы увидите опции для создания нового альбома, выбора настроек конфиденциальности и размера загружаемого изображения.



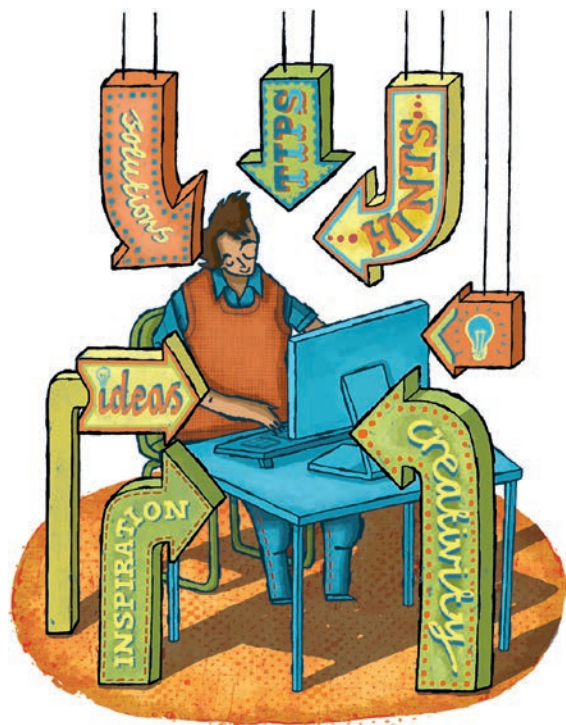
Доступ — всегда

Джонатан Робертс объясняет, как избежать катастрофы на рабочем месте, если отказал сервер, с помощью таких процессов, как балансировка нагрузки.



Наш эксперт

Джонатан Робертс сбежал из Башен *Linux Format*, чтобы попытаться счастья в качестве системного администратора.

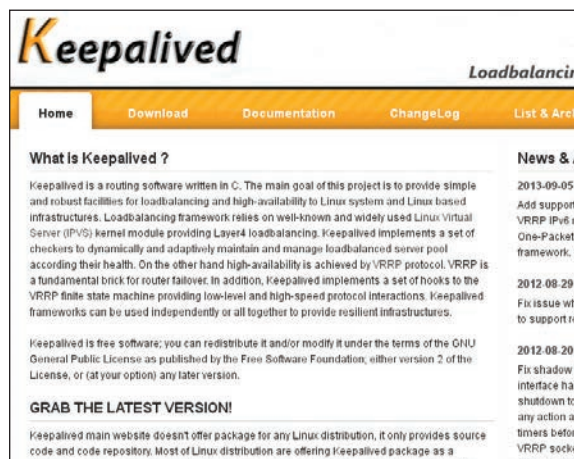


Если вы отвечаете за работу серверов на «производственном» участке, вероятно, что компания, организация или человек, для которых вы это делаете, полагается на сервисы, которые вы обеспечиваете. Они рассчитывают на них кругло-суточно, семь дней в неделю, 365 дней в году.

Однако вы живете в реальном мире и знаете, что компьютеры — вещь ненадежная. Во всех программах есть ошибки, и они всегда проявляются в самый неподходящий момент: приводят к отказу вашего почтового сервера именно в тот момент, когда ваш начальник отправляет жизненно важное сообщение; повреждают вашу базу данных, как только вы загрузили самого крупного клиента; или вся ваша сеть падает ровно в 9 утра, когда все сотрудники на местах и им не терпится приступить к работе.

Мы познакомим вас с несколькими простыми технологиями, которые позволят вам не обмануть тех, кто от вас зависит и ожидает непрерывной работоспособности, признавая при этом, что любое оборудование и любая программа может в некий момент отказать.

Независимо от технологии, основная идея везде одна: у вас есть несколько серверов для каждого типа сервисов, которые вы предоставляете — пять web-серверов, пара копий баз данных только для чтения (у вас также может быть и пара мастер-мастер для записи, однако при этом вы должны быть крайне осторожны, иначе у вас будут рассогласованные данные), и т.д. Если один сервер в кластере отказывает, это не страшно; есть другой, или даже несколько других, которые работают и доступ к которым продолжают иметь пользователи. Если один из ваших серверов



➤ **Keepalived** обеспечивает простое решение высокой отказоустойчивости для web-серверов и балансирования нагрузки, оставляя в стороне вопрос о курице и яйце.

отказал, вы теряете резерв, но ваши сервисы по-прежнему доступны, и вы можете чинить сломанный компонент в более спокойной обстановке.

Эта идея избыточности является ключевой для систем с высокой доступностью [high-availability, HA] — однако остается вопрос: как, собственно, реализовать такую систему?

Первая технология, с которой мы вас познакомим, именуется «балансировка нагрузки», и ее легко настроить для самых разных обстоятельств. Чтобы это продемонстрировать, возьмем самую простую ситуацию HA — web-сервер.

Два web-сервера лучше одного

При настройке web-сервера не останавливайтесь на одном; вместо этого настройте второй, и пусть у них будет одинаковые настройки и тот же контент. Если вам это кажется сложно, загляните в наше руководство по *Puppet* в **LXF174** и узнаете, как настроить идентичные серверы просто и надежно. При настройке вы, уж конечно, присвойте каждому статический IP-адрес, чтобы точно знать, куда обратиться для получения их контента.

В обычной ситуации вы также вводите эти IP-адреса в свою систему DNS, присваивая каждому серверу уникальное имя, например, `www1` и `www2`. Это все хорошо, но какой адрес вы дадите своим пользователям или клиентам — `www1.mycompany.com` или `www2.mycompany.com`? Если вас заботит HA, придется дать им оба адреса — если откажет первый, то это уже будет зависеть от пользователя, вспомнит ли он о втором сервере и примет ли решение использовать его.

Это не идеальный вариант: чем меньше вашим пользователям нужно думать о способах взаимодействия с серверами, тем лучше для всех заинтересованных лиц. Вы можете избежать этого двумя способами.

Первый именуется «карусельный DNS». Вместо того, чтобы присваивать серверам два разных имени в DNS, вы присваиваете

Злодейские SPOF'ы

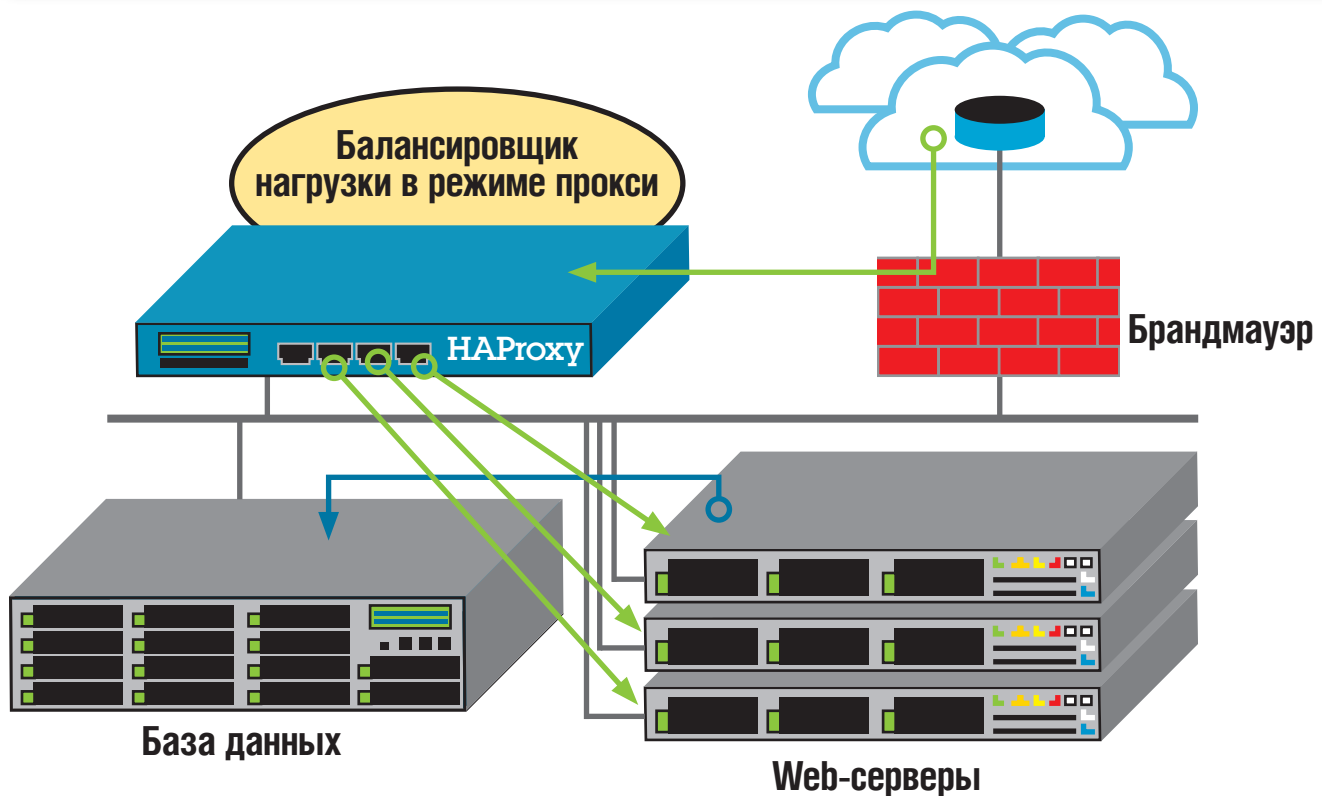
Претендент на награду за самый смешной акроним, SPOF'ы являют собой ключевую концепцию в среде HA. Это сокращение означает «единичная точка отказа [single point of failure]», и если вас заботит время пребывания в исправном состоянии, развейте на них нюх и искореняйте их, как только они появятся.

На самом деле, они говорят сами за себя: единичная точка отказа — это любая часть инфраструк-

туры, в которой отказ одного компонента может привести к отказу всей системы. Это может быть переключатель, брандмауэр или соединение Ethernet; хранилище, не настроенное в качестве массива RAID; или отдельная машина, предоставляющая важный сервис.

Инжиниринг вокруг SPOF иногда прост — например, приобретение добавочного жесткого диска для использования в массиве RAID 5; но иногда

решение бывает дорогостоящим, например, приобретение второго коммутатора и оснащение всех ваших серверов резервной картой Ethernet (включенной в конфигурацию сети). Ваши технические инстинкты должны подсказать вам необходимость устранения SPOF, как только они появляются, но будут ли они исправлены или нет, часто зависит от вашего руководителя и того, важен ли ему здоровый сон своего инженера.



то же имя обоим IP-адресам. Например, файл зоны может быть написан так:

```
www IN A 192.168.56.10
```

```
www IN A 192.168.56.12
```

Если вы используете DNS-сервер *Bind*, по умолчанию он работает с множественными данными вроде этих весьма своеобразным способом: все запросы на IP сервера `www` будут получать в ответ разные IP. И так, сначала Грэм ищет адрес `www.mycompany.com`, и получает в качестве первого адреса `192.168.56.10`; затем адрес запрашивает Эндрю, и получает `192.168.56.12`; когда адрес запрашивает Бен, он опять получает `192.168.56.10`.

Эта «карусель» ответов означает, что каждый сервер будет иметь дело с равными частями ваших пользователей. Если один из ваших серверов откажет, это, по крайней мере, повлияет только на 50% ваших пользователей! Будь у вас три сервера, получились бы только 33,33% пользователей, и т.д.

Однако и это не идеально. Для начала, если сервер откажет, часть ваших пользователей все же пострадает. И более того, в web-браузерах или других клиентах нет требования всегда посещать первый IP-адрес, возвращенный DNS сервером, так что эта

технология может вообще не сработать! И так, при всей ее простоте, все же, вероятно, она больше подходит для перераспределения нагрузки, чем для достижения HA.

HAProxy

Намного лучшим решением будет использование для балансировки нагрузки отдельного сервера. Выделенные балансировщики обеспечивают направление трафика на тот самый сервер, который вам нужен, и они могут совершать проверки работоспособности, чтобы трафик не направлялся на нерабочий сервер.

Балансировщики нагрузки работают, располагаясь между клиентом и сервером в прозрачном режиме. Вместо настройки записи DNS для `www.mycompany.com` на IP-адрес какого-то web-сервера, мы настраиваем DNS на указание IP-адреса балансировщика нагрузки. Весь трафик пойдет на балансировщик нагрузки, который будет заглядывать в каждый полученный пакет и выяснять, какой именно web-сервер должен его получить и ответить на него.

Балансировщики нагрузки могут принимать во внимание такие вещи, как исходный IP-адрес, порт назначения, «здоровье» »

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

серверов, на которые он может направить трафик, и многое другое.

Конечно, можно потратить кучу денег на коммерческий балансировщик нагрузки, но есть отличная опция с открытым кодом под названием *HAProxy*. Чтобы настроить *HAProxy* на работу с двумя web-серверами, первое, что вы должны сделать — установить его. В большинстве дистрибутивов это всего лишь означает использовать менеджер пакетов по умолчанию; на некоторых, например, на CentOS, сначала вам нужно установить дополнительный репозиторий, такой, как репозиторий Extra Packages for Enterprise Linux (EPEL).

Сделав это, вы должны отредактировать один файл настройки, чтобы он работал с вашими двумя web-серверами: `/etc/haproxy.conf`. В качестве простейшего примера, можете сделать следующую настройку:

```
frontend web_frontend *:80
    default_backend web_backend
backend web_backend
    mode http
    balance roundrobin
server www1 192.168.56.10:80 check
server www2 192.168.56.11:80 check
```



➤ *HAProxy* является свободным ПО с открытым кодом, которое выбирают желающие добиться гарантированно высокой отказоустойчивости.

```
server www2 192.168.56.11:80 check
```

Давайте разберем эту настройку.

Раздел интерфейса [frontend] говорит серверу *HAProxy*, какой порт и адрес слушать на предмет входящих соединений и на какую группу серверов эти соединения должны быть направлены. Он вводится ключевым словом **frontend**, затем ему дается имя, **web_frontend**, а затем перечисляются IP-адрес и порт — в данном случае все IP на порте 80.

Параметр **default_backend** в теле раздела указывает имеющуюся группу серверов движка, на которые направляются соединения.

Движок

Раздел движка являет собой то же самое: он вводится ключевым словом **backend**, ему дается имя, и затем набор параметров и значений устанавливаются в теле. В нашем примере мы сказали, что этот набор серверов работает в режиме HTTP (*HAProxy* также может работать как стандартный балансировщик нагрузки TCP), что для распределения запросов должен использоваться карусельный алгоритм, и затем мы переходим к определению наших двух узлов движка с помощью параметра `server`.

В операторе `server` мы также указали опцию `check`, которая велит *HAProxy* по умолчанию соединиться с web-сервером по указанному адресу и порту и проверить, получен ли успешный ответ, например, сообщение HTTP 200 OK. Если нет, он прекратит отправлять новые соединения на сбойный сервер, пока тот не пройдет тест.

HAProxy имеет массу опций настройки, который вы можете внести в этот файл, и у нас, естественно, нет места, чтобы задокументировать здесь их все — лучше заглянуть на их сайт: <http://haproxy.1wt.eu/download/1.5/doc/configuration.txt>. После этого вы должны суметь запустить *HAProxy*, командой `service` или скриптом `/etc/init.d`. При обращении к `www` вы должны увидеть свои соединения, перенаправленные на оба web-сервера.

RAID

Даже если вы разработали свою инфраструктуру так, чтобы избежать единичных точек отказа, ваша жизнь станет проще, если вы сможете избежать даже отказа сервера.

Одна из самых частых причин отказа сервера — жесткие диски. Это механические устройства, и они имеют ограниченный срок жизни — все жесткие диски когда-нибудь отказывают. Чтобы уменьшить риск отказа жесткого диска, вы можете настроить свое хранение в RAID (Redundant Array of Independent Disks — избыточный массив независимых дисков). RAID позволяет распределить данные по нескольким дискам, организовав данные в зависимости от того, надо ли вам повысить производительность или надежность. Эти настройки именуется «уровнями [levels]», и они таковы:

➤ **RAID 0** Располагает данные полосами на дисках поочередно. Это улучшает производительность, поскольку чтение и запись может производиться параллельно на нескольких дисках, однако отказ одного диска все равно приводит к отказу всего массива.

➤ **RAID 1** Зеркалирует данные на дисках. Прироста производительности нет, но если вы потеряете одну

сторону зеркала, другая продолжит работу, как и ваш сервер. Если вы хотите хранить 20 ГБ данных, вам понадобится 40 ГБ дисковой памяти.

➤ **RAID 5** Распределяет данные по дискам вместе с дополнительными метаданными, которые именуется информацией «контроля четности». Если отказывает один диск, эта информация может использоваться вместо потерянных данных, но производительность упадет. После замены поврежденного диска информацию о четности можно использовать для восстановления потерянных данных на новом диске, что восстановит производительность. Если отказывает более одного диска, восстановить информацию будет невозможно.

➤ **RAID 6** Похож на RAID 5, но поддерживает потерю двух дисков. Если на замену диска уйдет некоторое время или если диски слишком большие (и, следовательно, время восстановления слишком долгое), являет собой лучший выбор, чем RAID 5.

➤ **RAID 10** Полосы в зеркалах. То есть, зеркало диска с данными, расположенными полосами. Этот уровень предпринимает попытку предложить повышение производительности RAID 0 и резерв работоспособности RAID 1.

RAID 5 и 6 — наиболее частый выбор, если важна целостность и доступность данных. У них меньшая емкость хранения, чем у RAID 1 или RAID 10, но они ускоряют чтение. Иными словами, они медленнее записывают данные (потому что надо рассчитывать и записывать информацию о четности), и вы должны тщательно обдумать выбор уровня, в зависимости от того, что нужно вашему приложению.

Изучив уровни, вы должны их реализовать. Аппаратный способ — купить «устройство», карту-контроллер RAID, которая управляет массивом, рассчитывает информацию контроля четности и т.д., и ОС видит диски, как если бы это был один диск. Часто это сопровождается большим кэшем, что может помочь сделать массив быстрее.

Конечно, контроллеры оборудования бывают дороги, и вы можете предпочесть программный RAID. В Linux таковой управляется утилитой *mdadm* (multiple disk administration). Синтаксис прост:

```
mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sda /dev/sdb /dev/sdc
```

Этот пример создает новый массив RAID 5, отобразив как `/dev/md1` и состоящий из дисков `sda`, `sdb` и `sdc`.

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Балансировщики нагрузки HA

Мы изрядно продвинулись, введя в структуру *HAProxy* — по крайней мере, по сравнению с отсутствием резерва или карусельным DNS; однако теперь вы должны задаться вопросом: что будет, если рухнет сервер *HAProxy*? У вас будет два работающих, но бесполезных web-сервера, поскольку больше никому будет направлять на них трафик.

Изначально, если начать размышлять об этом вопросе, то у нас получится этакая проблема курицы и яйца — как можно ввести HA в уровень вашего балансировщика нагрузки, не прибавив при этом дополнительного уровня?

Самым простым ответом будет настроить «виртуальный» IP-адрес, разделяемый между двумя балансировщиками нагрузки. Да, верно, вместе с двумя web-серверами вам придется работать с парой балансировщиков нагрузки! Возможно, вам это начинает казаться несколько нерационально: чтобы обслуживать трафик с одного web-сервера, мы уже использовали три других сервера — web-сервер резерва, балансировщик нагрузки и web-сервер резерва. Поскольку требуются дополнительные ресурсы, вам нужно очень тщательно обдумывать, какую HA-технологии использовать.

Если вы выбираете HA просто для обслуживания вялого трафика, то балансировщик нагрузки будет перебором, и лучше тогда перепрыгнуть через уровень балансировщика нагрузки и сразу перейти к технологии виртуального IP, к объяснению которой мы как раз вплотную подошли. Однако если вам нужна HA и балансировка нагрузки между целыми кластерами серверов, то отдельный балансировщик с виртуальным IP будет для вас, вероятно, самым лучшим решением.

Виртуальные IP

Помня об этом предупреждении, давайте рассмотрим, что такое виртуальный IP и как его настроить.

Основная идея виртуального IP заключается в том, что помимо ваших двух серверов со своим обычным IP-адресом, третий IP-адрес управляется программой, работающей на каждом сервере, которые между собой обеспечивают ее «включение» только на одном из двух в любой момент времени.

Например, помимо того, что наши серверы имеют адреса 192.168.56.10 и 192.168.56.11, они будут делить еще и третий адрес — 192.168.56.12. Обычно мы настраиваем один в качестве первичного сервера-«мастера», а другой — в качестве резерва.

При нормальном режиме работы, сервер резерва проверяет, видит ли он сервер-мастер; то есть получает ли он ответ на отправленный ping. Стоит серверу резерва ответа не получить, как он переносит настроенный виртуальный IP на себя. Если мастер-сервер снова появляется в некий момент в будущем, он отказывается от виртуального IP, и мастер-сервер снова возьмет его на себя.

Чтобы все это использовать, вы настраиваете всю свою инфраструктуру, например, брандмауэры, чтобы трафик, предназначенный для ресурсов с HA, например, для наших web-серверов, направлялся на виртуальный IP-адрес. Если мастер отказывает, трафик автоматически перенаправляется на резервную машину.

Keepalived

Как и всегда, есть немало разных инструментов, с помощью которых вы можете реализовать виртуальный IP, но самый простой из них — *keepalived*. Чтобы его настроить, установите его, а затем поместите в */etc/keepalived/keepalived.conf* на обоих серверах что-то вроде

```

vrrp_instance VI_1 {
    interface eth0
    state MASTER
  
```

```

priority 101 # 100 on the backup box
virtual_router_id 51
virtual_ipaddress {
    192.168.56.12
}
}
  
```

Это настраивает экземпляр VI_1 «избыточного протокола виртуального роутера [virtual router redundancy protocol, VRRP]» с виртуальным IP-адресом 192.168.56.12. Если машина с большим приоритетом перестает отвечать, виртуальный IP будет перенесен на машину с меньшим приоритетом.

Keepalived можно расширить «проверочными скриптами», которые можно использовать для слежения за состоянием здоровья сервиса на web-сервере. Так, если вы применяете *keepalived* для поддержки балансировщика нагрузки HA, вы можете использовать следующее в *keepalived.conf*, чтобы проверить, работает ли процесс *HAProxy*:

```

vrrp_script chk_haproxy {
    script "killall -0 haproxy"
    interval 2
    weight 2
}
  
```

Этот скрипт каждые две секунды проверяет, работает ли *HAProxy*, и если да, то прибавляет два к приоритету.

Если у вас есть этот скрипт на обеих машинах, обе будут иметь большую приоритетность, пока *HAProxy* работает, и будут применяться обычные правила отказоустойчивости. Однако если *HAProxy* откажет на первой машине, у нее будет приоритетность на единицу меньше, чем на резерве, и ваш VIP-клиент автоматически переключится.

Информация к размышлению

В зависимости от того, какой сервис вы настраиваете для HA, стоит рассмотреть и другие технологии, каждая из которых хороша для определенных сервисов. Например, в прошлых **LXF** мы уже представили вам репликацию *MySQL*, но вам, возможно, нужно будет рассмотреть также и такие инструменты, как *DRBD* для репликации целых файловых систем, подчиненных DNS-серверов и прочего.

Возможностей слишком много, чтобы охватить их в одной статье, но многие из этих специальных технологий тем не менее должны сочетаться с балансировкой нагрузки или виртуальным IP, в той или другой форме. Если мне повезет, эта статья ознакомит вас с основными концепциями, который позволят вам создать инфраструктуру HA. **LXF**

► *Bind* большинству известен своим умением разрешать IP-адреса, но его также можно настроить на простейшую балансировку нагрузки. Однако он не столь надежен, как другие описанные здесь опции, поэтому применяйте его с осторожностью.

Место на диске:

Кэш браузера и временные файлы, оставшиеся после обновления, могут изрядно засорить компьютер. **Нейл Ботвик** показывает, как навести порядок.



Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не расскажет, где находится центральный сервер.

Жесткие диски нынче велики, дешевы и быстры, но сколько бы свободного места у вас ни было, оно имеет свойство исчезать. О да, это ISO-образы и фильмы, которые вы скачали, а может быть, вы даже пробовали какие-то дистрибутивы в *VirtualBox*; но есть и другие файлы, которые съедают свободное место — файлы, которые вы туда не помещали и о которых, возможно, даже не подозреваете. Что же это за файлы, где они находятся и как безопасно от них избавиться?

Самая частая причина скрытого использования диска — кэши. Когда программа обращается к файлу через сеть, она часто сначала сохраняет его локальную копию и потом работает с ней. Обычно это делается для кэширования, чтобы в следующий раз обратиться к файлу быстрее. Это активно делают браузеры, которые сохраняют копии изображений с web-страниц, чтобы в следующий раз при заходе на сайт снова их не загружать. Еще одна причина таких действий — проверка целостности файла. Менеджеры пакетов загружают пакеты в каталог кэша, а затем перед установкой проверяют, что контрольная сумма или подпись файла верны. Это позволяет защитить данные при передаче и, что важнее, защитить пакеты от подмены их содержимого злоумышленниками. Как же расходуется все это место? Команда **df** (disk free — свободное место на диске) показывает доступное и используемое место в файловой системе:

```
df -h /home
```

Но как узнать, где оно используется? Для этого есть несколько графических утилит, о которых мы поговорим позже, а стандартная команда — **du** (disk usage, использование диска).

```
du -sch ~
```

Она покажет общий объем, занимаемый домашним каталогом. При указании параметра **-h** размер выводится в человеко-читаемых единицах измерения, параметр **-s** суммирует содержимое каждого каталога вместо вывода списка его содержимого, а **-c** выводит в конце суммарный результат. Чтобы еще детальнее разбить отчет, воспользуйтесь командой

```
du -sch ~/*
```

Сравнив информацию об использовании диска от этих двух команд, вы заметите значительную разницу. Это различие обусловлено тем, как ***** интерпретируется оболочкой; она не включает скрытых файлов и каталогов, в которые входят такие каталоги, как **.cache** и **.config**. Чтобы в ***** включались файлы, начинающиеся на точку, нужно включить опцию **dotglob**.

```
shopt -s dotglob
```

Пользователи *Zsh* могут использовать в качестве такого шаблона ***(D)**. Так мы получим список файлов с указанием занимаемого места для каждого, но нам нужно выбрать самые большие; для этого мы перенаправляем вывод **du** команде **sort**.

```
du -s -BM * | sort -n | tail
```

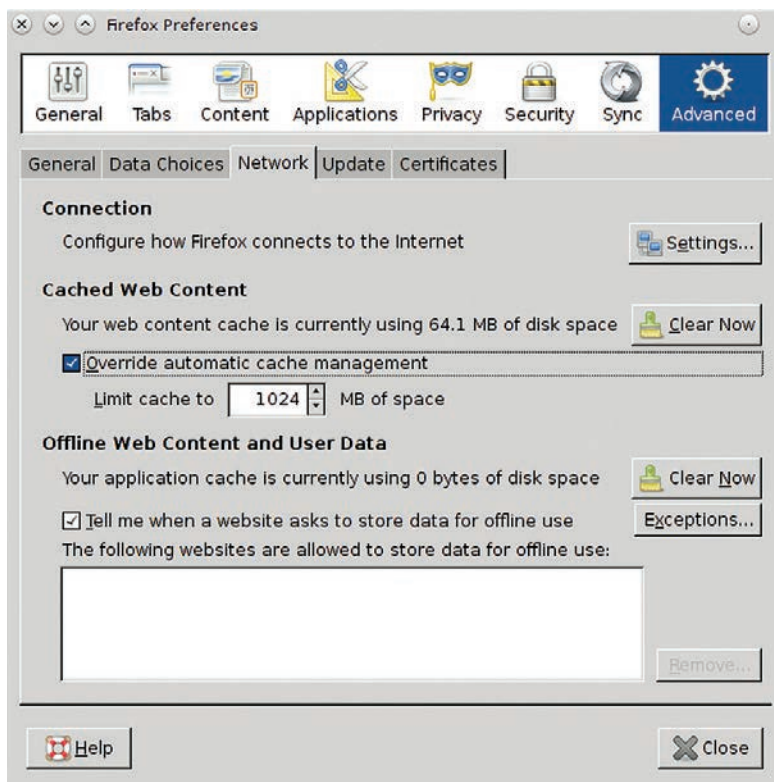
Параметр **-BM** команды **du** выводит все размеры в мегабайтах, параметр **-n** команды **sort** сортирует по числам, а **tail** показывает десять самых жадных каталогов. Эту команду можно выполнить на любом из этих каталогов, чтобы узнать, как используется место внутри каталога.

Раздутый браузер

Web-серфинг давно стал высокоэффективным способом тратить время, но он может заодно съесть гораздо больше места на диске, чем вы планировали. Изображения тоже кэшируются, и в этом есть смысл, но нужны ли вам на самом деле изображения с сайтов, на которые вы заходили год назад и вряд ли вернетесь снова? Если вы пользуетесь *Firefox*, размер кэша можно ограничить. Зайдите в Preferences > Advanced > Network [Настройки > Дополнительные > Сеть], поставьте галочку у *Override automatic cache management box* [Отключить автоматическое управление кэшем], и вы сможете задать максимальный размер кэша. Пользователям *Chrome/Chromium* повезло меньше: единственный способ ограничить размер кэша — запустить браузер с параметром **--disk-cache-size**, указывая размер кэша в мегабайтах.

Браузеры — не единственные программы, которым нужно много места под кэш. Во многих утилитах для просмотра изображений и файловых менеджерах есть просмотр миниатюр, и эти миниатюры кэшируются. Хотя их лучше иметь под рукой и не генерировать каждый раз, нужны ли вам миниатюры отпускных фото пятилетней давности?

При выводе содержимого каталога около каждого файла вы видите дату и время его последнего изменения. В Linux для каждого файла также хранится **“atime”**, время последнего доступа к файлу. В некоторых дистрибутивах это отключено, потому что немного влияет на скорость чтения файлов, особенно маленьких,



В *Firefox* можно ограничить место, используемое под кэш. По умолчанию браузер решает это сам.

Освобождаем

но нам оно пригодится. Эти кэшированные изображения могли быть созданы несколько лет назад, но если вы хотите удалить только те изображения, которые давно не использовались, вам и пригодится `atime`. Командой `find` можно найти все файлы, к которым не обращались, например, 90 дней:

```
find .cache -type f -atime +90
```

Найденные файлы можно удалить и в самой команде `find`:

```
find .cache -type f -atime +90 -exec rm "{}" +
```

Это зависит от того, включено ли в вашей файловой системе `atime` — что можно узнать из вывода команды `mount`. Если в списке опций для данной файловой системы нет `noatime`, то `atime` включено; в противном случае доступно только время изменения. Синтаксис команды `find` одинаков, за исключением того, что вместо `-atime` используется `-mtime`.

Если вы читаете почту через IMAP, то почтовый клиент сохраняет в кэше копию каждого письма после его прочтения. Со временем кэш разрастается, храня и письма, которые вы вряд ли прочтете снова. Но помните, что после удаления писем из кэша они загрузятся туда снова, в следующий раз, когда вы воспользуетесь поиском в почтовом клиенте; поэтому очищайте кэш, только если места совсем мало (или вы вообще не пользуетесь поиском).

Это не только home

Кроме кэша в домашнем каталоге, кэшем может пользоваться и файловый менеджер при установке обновлений или новых программ. Местоположение этих файлов и то, можно ли их безопасно удалить, зависит от менеджера пакетов. Во всех дистрибутивах на основе Debian или Ubuntu используется менеджер `APT` (центр управления программами и `Synaptic` — просто дружелюбные оболочки `APT`). `APT` хранит копии загруженных пакетов в `/var/cache/apt`. Удалять их вручную не нужно: у `APT` есть параметры для очистки кэша. В активно используемой и часто обновляемой системе кэш пакетов может занимать довольно много места на диске. Основная команда для очистки кэша — `clean`:

```
apt-get clean
```

Она удаляет все загруженные пакеты. Также имеется команда `autoclean`, которая удаляет только те пакеты, которых нет в репозиториях, то есть ненужный хлам. Каждую из этих команд можно запустить с параметром `--dry-run`, чтобы посмотреть, какие файлы будут удалены, не удаляя их на самом деле.

RPM тоже хранит загруженные пакеты в `/var/cache`, хотя точное местоположение зависит от менеджера пакетов. `Yum` в Fedora использует примерно такой путь: `/var/cache/yum/x86_64/19/fedora/packages`, но точный путь зависит от используемой версии. Отсортировав файлы по версии релиза, можно легко избавиться от старых пакетов, оставшихся после обновления с предыдущей версии. Для `Yum` указанной выше команде эквивалентна

```
yum clean all
```

Она удаляет из кэша все загруженные пакеты и другие кэшированные данные. База данных RPM не затрагивается. Также есть команда

```
yum clean packages
```

которая удаляет файлы уже не установленных пакетов.


В openSUSE по умолчанию используется менеджер пакетов `zypp`, который хранит файлы в `/var/cache/zypp`, но, в отличие от других дистрибутивов, по умолчанию удаляет файлы после установки, так что с ним ничего удалять не нужно. Пользователи

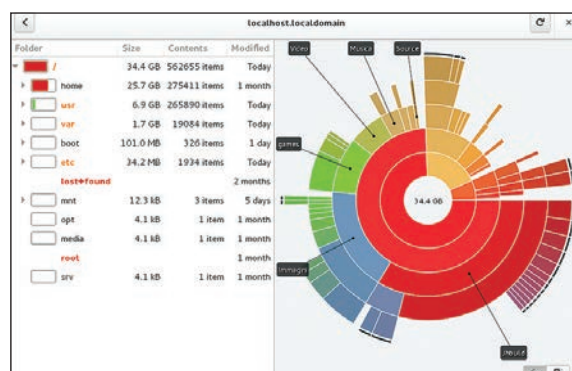
Gentoo могут удалить файлы из каталога `distfiles` командой `eclean-dist`. Все описанные здесь команды нужно запускать либо от имени `root`, либо с `sudo`.


Пока мы пользовались только утилитами командной строки. Для этого есть несколько причин: их можно найти во всех дистрибутивах (большинство из них входят в пакет `coreutils`); ими можно пользоваться даже тогда, когда из-за забитого диска вы не можете загрузить рабочий стол; и ими можно пользоваться удаленно. Есть и графические программы, и они особенно удобны тем, что умеют визуализировать использование пространства на диске. `Filelight` — программа для KDE, которая отображает потребление дисковой памяти в виде своего рода вложенной круговой диаграммы, по которой можно перемещаться и находить файлы и каталоги, которые используют больше всего места, и открывать в файловом менеджере те, которые нужно изучить более детально. Да, для KDE есть программы и без буквы K в названии! Эквивалент для Gnome называется `baobab`, хотя обычно, к счастью, его называют `Gnome Disk Usage Analyser` [анализатор использования диска Gnome].

Эти две программы являются графическим эквивалентом `du`, и есть как минимум еще одна графическая программа для удаления файлов и освобождения места на диске — `Bleachbit` (<http://bleachbit.sourceforge.net>). `Bleachbit` работает в домашнем каталоге и знает о том, какие типы файлов используются различными программами и рабочими столами. В ней можно просмотреть список, выбрать те области, которые нужно очистить, и нажать Preview [Предпросмотр], чтобы увидеть, сколько места будет освобождено. Быстрый запуск программы на моем компьютере показал, что только в `home` можно освободить более 2 Гб.

Не будьте слишком рьяными — очистка кэша браузера может освободить много места, но после удаления всех куки и сохраненных паролей вам придется вводить их снова на каждом сайте, на который зайдете. В окне Preferences [Настройки] есть несколько полезных опций, например, белый список каталогов, которые никогда не нужно очищать, и даже возможность перезаписать файлы перед их удалением. Однако настоящие параноики (а ныне это большинство из нас), наверное, воспользуются `shred`, чтобы окончательно избавиться от содержимого файлов. Помните, что `Bleachbit` предназначен для очистки домашнего каталога. С такими вещами, как кэш менеджера пакетов, лучше всего работать с терминала.

Ну вот, вы освободили много места на диске — можно загрузить гораздо больше новых файлов и захлестить его снова! 



 `Filelight` и `Gnome Disk Usage Analyser` дают графическое представление о потреблении дисковой памяти.

Рабочий стол:

Да, вы можете создать свой личный рабочий стол с низким потреблением ресурсов, не написав ни строки кода. Учитесь у **Майка Сондерса!**



Наш эксперт

Майк Сондерс начал использовать Linux с *FVWM* (в дни Red Hat 5.1), провел несколько лет с *Window Maker*, и теперь чувствует себя здесь как дома.



Строки заголовков, программы запуска приложений, панели, области уведомления, фоновые изображения — всего этого мы ждем от современного рабочего стола. Обычно эти функции предоставляются отдельными программами:

если вы работаете в *Xfce* и, например, введете в терминале `ps ax`, вы увидите `xfwm` для менеджера окон, `xfce4-panel` для обеспечения экранных панелей, и т. д. Если вы начнете выключать эти программы, части вашего рабочего стола примутся исчезать с экрана. KDE и Gnome работают примерно так же.

Что ж, все это хорошо и мило, но мы же линуксоиды, верно? Настройки по умолчанию работают нормально, но ведь мы используем эту ОС так, чтобы мы сами могли настроить ее по своему усмотрению, вырезать части, которые нам не нравятся, и исследовать внутреннее устройство системы. KDE, Gnome и *Xfce* выглядят, как жутко сложные монстры, однако на самом деле они — ничто иное, как набор программ, запускаемых в определенной последовательности. Вы можете заменять эти программы по отдельности или выбрать собственную комбинацию менеджера окон, панелей, переключателя рабочих пространств, и так далее — этим-то мы и займемся на нашем уроке.

Полным-полно легковесных менеджеров окон, предлагающих только голый минимум функций, поэтому другие разработчики написали дополнительные компоненты, нужные вам для создания полнофункциональной среды рабочего стола. Мы рассмотрим разные возможности, а затем покажем вам, как собрать их воедино в индивидуальный рабочий стол специально для вас. Если вы работаете в KDE, Gnome или *Xfce*, вы увидите, что в результате получится куда более быстрая и менее требовательная к памяти работа.

Однако самое важное — то, что теперь, услышав от своих братьев по Linux, что они обновили свой KDE или Gnome до самой последней версии, вы можете пренебрежительно махнуть рукой и сказать: «Фи! Отстой. Я вчера сделал собственный рабочий стол — могу и вам установить, если хотите». И вот уже ваш рейтинг умника взлетел до небес...

Выберем менеджер окон

➤ **Добавление менеджеров окон через менеджер пакетов должно создать новую сессию экрана приглашения — но мы опять-таки покажем вам, как это работает.**

Пожалуй, важнейший компонент — это менеджер окон (ОМ). Он (обычно) предоставляет строки заголовков, чтобы вы могли перемещать окна по экрану, а также кнопки для выполнения всяких действий с окнами — максимизировать, закрыть и т. д. Кроме того, ОМ (обычно) добавляет рукоятки к краям окон, позволяя менять их размер. Без ОМ ваши приложения вполне обойдутся, но вы не сможете перемещать их окна или изменять их размер. Есть несколько новых менеджеров окон, которые практикуют другие подходы, например, автоматически привязывают окна к сетке (плиточные ОМ), и им не нужны строки заголовков.

Здесь мы опишем несколько лучших вариантов и поясним, как их использовать. Если вы установите их через менеджер пакетов своего дистрибутива, у вас должно получиться выйти из текущей сессии и выбрать их в окне приглашения. Если вы не видите опции для своего любимого ОМ, попробуйте выбрать

опцию `fallback/terminal` в окне приглашения; в некоторых дистрибутивах она есть. Вы увидите пустое окно терминала, где можно ввести нужную команду, чтобы запустить ОМ, как описано ниже. Если вам вообще не найти ОМ для запуска, не волнуйтесь: далее мы покажем вам, как создать новую сессию экрана приглашения.

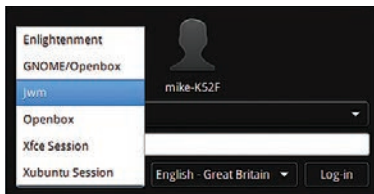
➤ **Название:** *JWM*

➤ **Сайт:** <http://joewing.net/projects/jwm/>

➤ **Команда для запуска:** `jwm`

JWM весьма минималистичен: он написан на C, и все его зависимости — только основные библиотеки X. Следовательно, он невероятно быстр. У *JWM* довольно удобная панель по умолчанию, с меню программ, переключателем рабочих областей и панелью задач, но, вероятно, вам понадобится более наполненная функциями панель из опций, которые мы рассмотрим позднее.

По умолчанию меню программ (доступное также по щелчку левой кнопкой мыши на рабочем столе) содержит только пункт для запуска терминала, блокировки экрана и выхода — вы можете настроить его с помощью редактирования `/etc/jwm/system`.



Сделаем сами

`jwmrc` (или `/usr/local/etc/jwm/system.jwmrc`, если вы компилируете его из исходника).

Этот файл настройки на основе XML прост для понимания, и с его помощью вы также можете настроить шрифты и привязку клавиш ОМ.

» **Название:** *Openbox*

» **Сайт:** www.openbox.org

» **Команда для запуска:** `openbox`

Openbox — часть семейства менеджеров окон *box, включающего *Fluxbox* и *Blackbox* (оригинал, от которого и произошли все остальные). Это, вероятно, самый популярный, не отличающийся лишним весом менеджер окон, и он используется на рабочем столе *LXDE*. Настройка по умолчанию *Openbox* крайне проста: запустив его, вы увидите перед собой полное ничто. Только курсор мыши в центре экрана.

Но... щелкните правой кнопкой, и появится небольшое меню, из которого можно открыть окно терминала и запустить веб-браузер или инструмент настройки. Именно этот инструмент и делает *Openbox* нашим любимым выбором как индивидуальной среды рабочего стола, потому что в менеджере окон настраивается практически все, без необходимости тыкать по текстовым файлам.

Плюс к тому, хотя *Openbox* очень ограничен в плане функций, он отличается высоким соответствием стандартам и отлично взаимодействует с другими инструментами.

Горячие клавиши типовые: `Alt+Tab` для перемещения по окнам; `Alt+F4`, чтобы закрыть программу; и `Ctrl+Alt`+стрелка вправо/влево для переключения между разными виртуальными рабочими пространствами.

» **Название:** *Ratpoison*

» **Сайт:** <http://ratpoison.wxcvbn.org>

» **Команда для запуска:** `ratpoison`

А вот совершенно иной вид менеджера окон: он использует концепцию, именуемую «мозаикой». Вместо обычной системы окон, которые можно перетаскивать и которые могут перекрывать друг друга, в *Ratpoison* окна привязаны к определенным областям экрана. Чтобы к этому привыкнуть, нужно некоторое время; но, запомнив горячие клавиши для управления окнами, вы будете все меньше и меньше зависеть от мыши (отсюда и название [*Ratpoison* — *англ.* «крысиный яд»]).

При первом запуске *Ratpoison* совершенно пуст, и вы увидите, что мышь ничего не делает. Нажмите на `Ctrl+t`, а затем `!`, и в верхнем правом углу появится крошечное окно, которое позволит вам ввести команду (например, `xterm`). Когда вы запусти-

Разговор на одном языке

Все эти инструменты могут работать вместе отнюдь не случайно. Linux критикуют за то, что это якобы крошка из старых и новых технологий, слабо связанных между собой, но это несправедливо. Конечно, старый код пока еще тут, но благодаря соблюдению стандартов программы отлично ладят друг с другом.

EWMH, спецификация Extended Window Manager Hints, «определяет взаимодействия между менеджерами окон, приложениями и утилитами, которые формируют часть среды рабочего стола». Иными словами, он позволяет этим компонентам делиться информацией, предоставляя

более связную работу. Хороший пример — переключатели рабочих областей: они часто показывают миниатюрные эскизы рабочих областей и окна в них. Обычно в этих эскизах вам нужно видеть только окна приложений, а не обычную фурнитуру рабочего стола. Благодаря EWMH, панель или менеджер окон может сообщить пейджеру, что окно не предназначено для отображения в эскизах. Загляните в <http://standards.freedesktop.org/wm-spec/wm-spec-latest.html>, где вы найдете полную спецификацию — это чтение долгое, но непременно, если вас интересует написание собственного ОМ.

те программу, по умолчанию она появится в полноэкранном режиме; открыв несколько окон, вы сможете переключаться между ними с помощью `Ctrl+t` плюс `n` или `p` (`next` — следующее и `previous` — предыдущее). Введите `Ctrl+t w` — отобразится нумерованный список окон, а по `Ctrl+t` и затем номеру вы перейдете к конкретному окну. И теперь в режиме мозаики начинается настоящая потеха. Открыв несколько окон, нажмите на `Ctrl+t`, чтобы разделить экран по вертикали на две части — их называют фреймами. Текущее окно появится в верхнем фрейме, а предыдущее окно — внизу. Если нажать `Ctrl+t`, а затем `S` (заглавную), то экран поделится по горизонтали. Это отлично, если у вас большой монитор с высоким разрешением, поскольку вы сможете, например, создать большой фрейм для вашего веб-браузера, и рядом с ним будет еще парочка меньших окон терминала.

Чтобы переключаться между фреймами, нажмите на `Ctrl+t`, затем клавиши курсора (например, стрелка вниз — для перехода на нижний фрейм). Вы можете сделать фрейм на полный экран снова с помощью `Ctrl+t Q`. Чтобы посмотреть полный список горячих клавиш, введите `Ctrl+t ?`, и загляните в *wiki Ratpoison* по адресу <http://tinyurl.com/obcarv5>, где вы увидите подробности по остальным функциям в ОМ.

Другие менеджеры окон, достойные рассмотрения, включают *Sawfish* (<http://sawfish.wikia.com/wiki/>), *WindowLab* (<http://nick-gravgaard.com/windowlab/>) и *Pekwm* (www.pekwm.org).

Выберем панель

Итак, работа по управлению окнами обеспечена. Теперь нужна панель для запуска программ, переключения между окнами и рабочими пространствами, отображения часов и так далее. Как уже упоминалось, *JWM* содержит простую панель, но мы здесь рассмотрим опции с большим количеством функций.

» **Название:** *Docky*

» **Сайт:** www.go-docky.com

» **Команда для запуска:** `docky`

Нам безумно нравится описание на сайте *Docky*: «Если в истории человечества и было создано нечто действительно потрясающее, то имя ему — *Docky*». И знаете, мы с этим почти согласились: он и вправду впечатляет. *Docky* довольно тяжел для панели, у него много зависимостей, и если вы создаете рабочий стол для машин со скромным объемом памяти, лучше будет выбрать другую опцию. *Docky* должен найтись в репозиториях пакетов большинства дистрибутивов (в дистрибутивах на базе Ubuntu ему нужно всего

лишь `sudo apt-get install docky`); запустив его, вы увидите черную панель внизу экрана. На ней один-единственный статический значок с якорем — для открытия диалогового окна `Settings`. Отсюда можно выбрать режим `Panel`, который расширяет `Docky` на весь экран, и изменить тему (есть очень красивые). Переключитесь во вкладку `Docklets` диалогового окна `Settings`, чтобы добавить в панель другие статические объекты, например, часы, переключатель рабочих областей и монитор заряда батареи.

У `Docky` нет основного меню, что несколько сдерживает возможности, но, запуская программы из менеджера окон, вы увидите, как на панели возникают их значки. Щелкните правой кнопкой по значку и выберите “Pin to dock [Прикрепить к панели]”, чтобы значок оставался на панели даже после закрытия программы. В итоге вы заселите панель значками тех программ, которыми пользуетесь чаще всего, и отсутствие меню не будет трагедией.

» **Название:** *Cairo-Dock*

» **Сайт:** <http://glx-dock.org>

» **Команда для запуска:** `cairo-dock`

Тем, кто ищет бальзам для глаз, *Cairo-Dock* предлагает его в избытке. Его вид “3D Plane” невероятно похож на док Mac OS X, с эффектами отражения и линиями раздела областей. Если навести мышью на значок, тот увеличится в размере; а если на папку, на экране появится ее содержание. Все в целом очень красиво — но, конечно, вашей видеокарте придется попотеть.

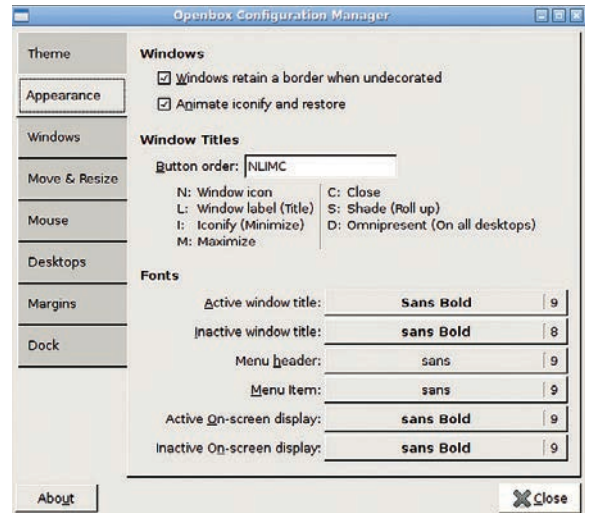
В отличие от `Docky`, в *Cairo-Dock* основное меню есть — на основе системных настроек; иными словами, оно будет показывать те же объекты, какие вы видите в других менеджерах окон и средах рабочего стола. Можно щелкнуть правой кнопкой по любому объекту, чтобы его настроить, или перейти в `Cairo-Dock > Configure`, чтобы настроить саму его панель. А настройке он поддается очень хорошо: здесь масса включаемых дополнений, переопределяемых горячих клавиш и изменяемые местоположение и размер.

» **Название:** *WBar*

» **Сайт:** <http://code.google.com/p/wbar/>

» **Команда для запуска:** `wbar`

Это самая легковесная из трех представленных здесь панелей; она запускается почти мгновенно. В ней прозрачный фон,



» *Openbox* имеет собственный инструмент настройки (*obconf*), который устраняет необходимость возиться с текстовыми файлами.

анимированные значки, увеличивающиеся при наведении на них курсора мыши, и графический инструмент настройки (установите `wbar-config` вместе с самой панелью). По умолчанию она отображается вертикально, но вы можете щелкнуть по самому верхнему значку, чтобы изменить и это, и другие настройки.

Добавляйте и удаляйте программы запуска через вкладку `Icons`, а если вы хотите, чтобы панель работала скорее как панель задач, опцию для этого вы найдете во вкладке `Preferences` [Параметры]. Уровни изменения масштаба, эффекты прозрачности и расстояния между значками настраиваются во вкладке `Effects`.

Если ни одна из этих панелей вас не прельстила, попробуйте *tint2* (<http://code.google.com/p/tint2/>), *PerlPanel* (<http://savannah.nongnu.org/projects/perlpanel>) или *Avant Window Navigator* (<https://launchpad.net/awn>). Последние две на данный момент уже малость устарели, но вы все равно сможете найти пакеты для разных дистрибутивов в Сети.

Выберем дополнения

Большинство легковесных менеджеров окон не предусматривают возможности настроить фоновое изображение экрана, и для этого вам понадобится другой инструмент. *Hsetroot* — простенькая программка, имеющаяся в большинстве дистрибутивов в виде отдельного пакета, а иногда в качестве части *HackedBox* (еще одного небольшого ОМ). Стоит вам его установить, пользоваться им будет крайне просто:

```
hsetroot -full /path/to/image.jpg
```

Это поместит полное изображение на экран, без всяких изменений. Среди прочих опций — `-center`, `-tile` и `-fill` (чтобы его растянуть). Можете также использовать изображения в формате PNG, да и в других форматах.

Если выбранная вами панель не предусматривает переключатель виртуальных рабочих областей, вам нужен отдельный переключатель. В этой области выбор невелик, но мы рекомендуем

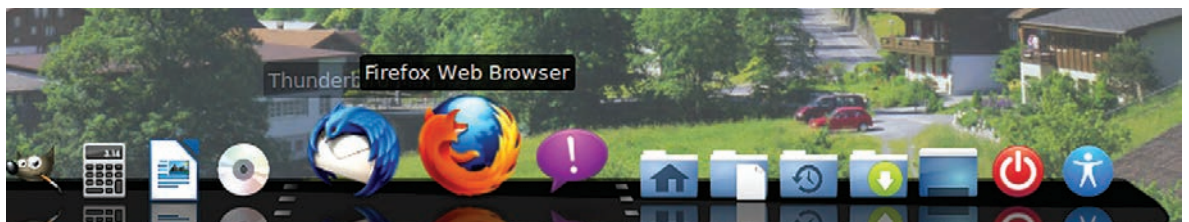
(поскольку его зависимости минимальны и он работает практически везде) *NetWMPager*. Зайдите на <http://sourceforge.net/projects/sf-xpaint/files/netwmpager/> и скачайте самый последний исходный код (на момент написания данной статьи это был 2.04) и скомпилируйте его так:

```
tar xfv netwmpager-2.04.tar.bz2
cd netwmpager-2.04/
./configure
make
sudo make install
```

Если столкнетесь с проблемой «нераспознанная ссылка на символ `XgetWindowAttributes` [undefined reference to symbol `XgetWindowAttributes`]» после шага `make`, отредактируйте `config.mk`, изменив строку `XFT_LIBS`, чтобы она читалась так:

```
XFT_LIBS = -lXft -lX11
```

» Хотите добавить Mac OS X-подобных прелестей на свой рабочий стол? *Cairo-Dock* — то, что вам нужно.



Сохраните файл и повторите два последних шага еще раз. Теперь можете запускать его с помощью **netwmpager**, и вы увидите окно программы листания, отображающее рабочие столы в нижнем левом углу.

По умолчанию оно очень маленькое, и автоматически скрывается; чтобы это исправить, отредактируйте **/usr/local/share/netwmpager/config-example**, изменив строки **geometry** и **auto_hide**. Если вам нужна область уведомлений (она же системный лоток), отличным примером будет *Stalonetray*. Он есть в архивах пакетов

Ubuntu и в виде исходного кода на <http://stalonetray.sf.net> — если будете компилировать из исходника, распакуйте его и пройдите стадии **.configure**, **make** и **sudo make install**, как описано выше.

Потом запустите его так:

```
stalonetray -geometry 5x1+50+20
```

Создается область уведомлений размером пять слотов в ширину и один слот в высоту, 50 пикселей в ширину и 20 вниз (с левого угла экрана). Конечно, вы можете сделать ее любого размера и разместить в любом месте по своему усмотрению.

Соберем все это вместе

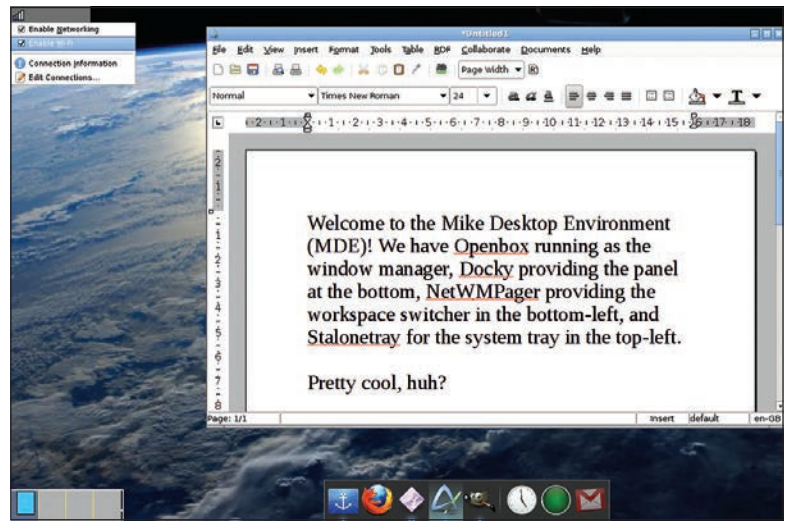
Теперь мы готовы собрать все эти компоненты вместе и создать среду рабочего стола. Вы можете использовать любую комбинацию из вышеупомянутых программ; в нашем случае мы возьмем *Openbox*, *Docky*, *hsetroot*, *NetWMPager* и *Stalonetray*. Первым делом надо создать скрипт, запускающий нужные программы. Для нашего индивидуального рабочего стола мы поместим его в **/usr/local/bin/mikedesk-start**:

```
#!/bin/bash
docky &
hsetroot -fill /usr/share/backgrounds/space-02.jpg &
netwmpager &
stalonetray -geometry 5x1+0+0 &
openbox
```

& после команды здесь означает «запустить их в фоновом режиме» — т.е. они все могут работать в одно и то же время, и не ждут, пока другие программы закончат работу. Важно, чтобы ваш менеджер окон шел в последней строке, без **&**, иначе сессия немедленно закроется.

Сохраните файл и сделайте его исполняемым (например, **sudo chmod +x mikedesk-start**). Следующий шаг — создать сессию экрана приглашения, поэтому создайте новый файл с названием наподобие **/usr/share/xsessions/mikedesk.session** и следующим содержимым, настроенным индивидуально для вашей среды:

```
[Desktop Entry]
Name=MikeDesk
Comment=Mike Desktop Environment
Type=XSession
```



➤ А вот и результат нашей работы: среда рабочего стола, созданная с нашим собственным выбором компонентов.

```
Exec=/usr/local/bin/mikedesk-start
```

```
TryExec=mikedesk-start
```

Сохраните его, выйдите из текущей сессии, и вы должны увидеть новую сессию в своем менеджере приглашений. Выберите ее, войдите и наслаждайтесь плодами своих трудов! Поздравляем: теперь вы официально подтвердили свою крутизну. **LXF**

Помогите! Мои приложения GTK и Qt стали уродами

Увы, это весьма частая проблема при переходе на другой менеджер окон. Вы идеально все настраиваете, устанавливаете превосходные темы, выбираете чудесные обои рабочего стола, а потом... Ваши приложения *GTK* и *Qt* выглядят, как убогое старье из 1980-х. Плоские меню, угловатые виджеты, темно-серый фон и уродливые шрифты — какая-то скверная штука.

GTK и *Qt* больше не могут определять, какую тему использовать, поэтому они используют уродливые встроенные темы по умолчанию. Если вы работаете в *Gnome* или *Xfce*, приложения *GTK* могут попросить рабочий стол предоставить информацию по темам, но в отдельном менеджере окон эти наборы инструментов вынуждены полагаться на файлы настройки. Для *GTK* есть несколько опций; наилучшим выходом будет попробовать их комбинировать. *gtk-chtheme* позволяет настроить тему,

используемую приложениями *GTK* 2.x, но не 3.0. Для последних отредактируйте **~/.config/gtk-3.0/settings.ini** чтобы он указывал имя темы:

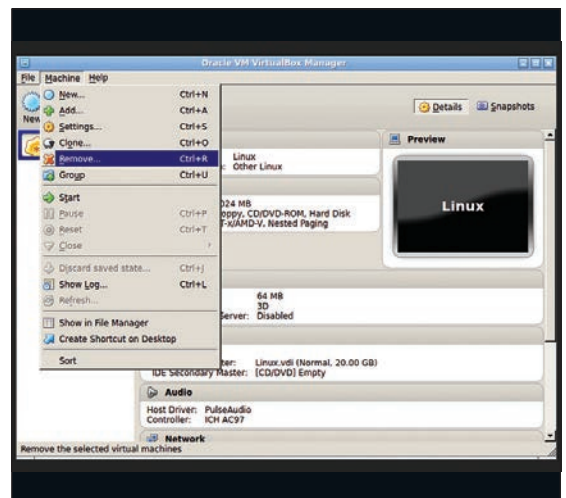
```
[Settings]
gtk-theme-name = Clearlooks
gtk-fallback-icon-theme = gnome
```

Если ничто из этого не произвело желаемого эффекта и у вас установлен *Gnome*, попробуйте запустить **gnome-settings-daemon**. Это даст новые темы работающим приложениям *GTK*, используя настройки *Gnome*, но ваши обои рабочего стола изменятся.

Для *Qt* есть две опции: чтобы заставить приложения *Qt* использовать темы *GTK*, отредактируйте **~/.config/Trolltech.conf** и добавьте это:

```
style=GTK+
```

Альтернатива — установить пакет **qt4-gtkconfig** и затем запустить **qtconfig**.



➤ *VirtualBox* (приложение *Qt*) с темой *Qt* по умолчанию выглядит вылитым *Windows 95*.

SSH. На несколько

Нейл Ботвик показывает, как выполнять команды сразу на нескольких компьютерах, разнесенных географически.



Наш эксперт

У Неила Ботвика по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не расскажет, где находится центральный сервер.

SSH — фантастическое средство для работы на удаленном компьютере или его администрирования, независимо от того, находится этот компьютер на другом конце света или в другом углу комнаты. Открыв оболочку на другом компьютере, на нем можно работать, а при необходимости — открывать дополнительные терминалы или дополнительные вкладки в том же терминале; правда, чем больше компьютеров, тем менее это удобно. Если компьютеров больше двух-трех, и особенно если вы выполняете на них одну и ту же задачу, набирать одни и те же команды снова и снова становится скучно, и, что гораздо важнее, так проще допустить ошибки. Существует несколько программ, позволяющих выполнять команды в нескольких SSH-сеансах одновременно; мы рассмотрим две — **dsh** и **cssh**.

Распределенные оболочки

Dancer's Shell [Оболочка танцора], или **dsh**, выполняет одну и ту же команду на нескольких компьютерах, по умолчанию — через SSH. Результаты всех команд выводятся в тот терминал, где запускалась команда **dsh**. Например:

```
dsh -m hactar -m root@marvin uptime
root@marvin: 10:05:20 up 8 days, 24 min,
hactar: 10:05:22 up 2 days, 18:41,
```

Здесь указанная команда запускается на каждом из компьютеров, указанных с параметром **-m** (или **--machine**). К именам компьютеров применяется стандартный синтаксис SSH, поэтому к первому компьютеру мы подключаемся от имени пользователя, который запускал **dsh**, а к второму — от имени **root**. Каждая строка вывода предваряется именем компьютера, и вы видите, что откуда пришло. Это удобно, если команда выводит всего одну строку; ну, а если, например, вывести список содержимого каталога? Команда

```
dsh -m host1 -m host2 -m host3 ls -l | less
```

выводит содержимое домашнего каталога на каждом компьютере, но результаты отдельных команд могут перемешаться, поскольку **dsh** просто выводит части результатов отдельных команд сразу после их получения. Чтобы это обойти, можно воспользоваться параметром оболочки **-w**, или **--wait**. С этим параметром команда выполняется на каждом удаленном компьютере по очереди, и подключение к следующему компьютеру производится только после того, как команда на предыдущем закончила работу.

Это позволяет получить аккуратные результаты за счет снижения скорости. Если компьютер недоступен, то перед переходом к следующему **dsh** будет ждать, пока подключение завершится неудачей, так что данным способом стоит пользоваться лишь тогда, когда это в самом деле необходимо. Если вы будете пользоваться **dsh**, настройте в сеансах SSH между компьютерами аутентификацию по ключу (это следует сделать в любом случае по соображениям безопасности), иначе придется набирать пароль при каждом запуске команды.

Создание групп

Если набирать список имен компьютеров, это быстро наскучит, поэтому **dsh** поддерживает группы. Они задаются в файле **/etc/dsh/group** или в файле **~/dsh/group**. Имя файла — это имя группы, и он содержит список хостов (с именами пользователей или без) по одному на строку, например:

```
cat ~/dsh/groups/homelan
oolon
root@lunkwill
marvin
```

Теперь вместо того, чтобы каждый раз набирать имена хостов, можно посмотреть, какой пользователь зашел на каждый из компьютеров, командой

```
dsh -w -g homelan w
```

Также есть специальная группа в файле **/etc/dsh/machines.list** или **~/dsh/machines.list**, она используется с параметром **-a**, или **--all**, что позволяет даже не указывать имя группы:

```
dsh -a somecommand
```

При настройке групп вы можете обратить внимание на скорость отклика разных компьютеров и их доступность. Если вы пользуетесь параметром **-w**, лучше всего поместить самые быстрые компьютеры в начале, а самые ненадежные — в конце списка, чтобы сэкономить на ожидании. Поведение программы можно настроить в конфигурационном файле **/etc/dsh/dsh.conf**. Настройки по умолчанию, не считая параметра **-w**, обычно отлично подходят для подключения по SSH. Я обычно пользуюсь такими:

```
remoteshell = ssh
showmachinenames = 1
waitshell=0
```

Параметры команды **dsh** обрабатываются текущей оболочкой, поэтому специальные символы нужно экранировать — * станет * и т.д. Вывод команд со всех хостов тоже проходит через **dsh**, поэтому команда

```
dsh -a echo "some text" >somefile
```

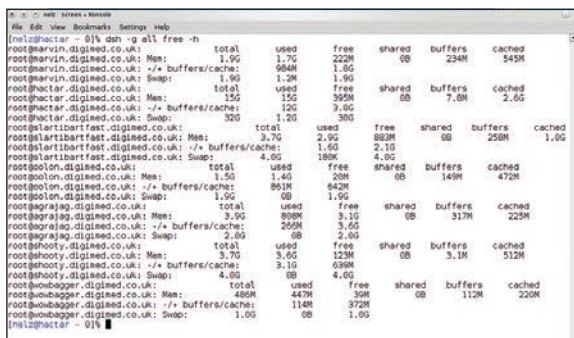
поместит весь вывод в файл в текущем каталоге, так как перенаправление работает для процесса **dsh**. Взамен можно попробовать нечто вроде

```
echo "some text" | dsh -a tee somefile
```

так как стандартный ввод будет пропущен через **dsh** к каждой из удаленных оболочек.

Хотя мы рассмотрели с **dsh** в основном команды для вывода информации, так как с ними получаются более ясные примеры, вы можете выполнять на удаленных хостах любые команды,

➤ Запуская одну и ту же команду на нескольких компьютерах с **dsh**, не делайте этого с командами, у которых большой объем выводимых данных.



СИСТЕМ

Screen

ClusterSSH по-настоящему вступает в свои права вместе с утилитой *screen*, поэтому одно из первых действий при настройке — создать пункт меню Send для команды **screen -xRR**. Если вы никогда не пользовались *screen* (или *tmux*, которая выполняет похожие действия), попробуйте.

Screen — оконный менеджер для терминалов, который позволяет запустить более одного сеанса в одном и том же терминале, но это только начало. Наберите

screen в терминале, и на первый взгляд не произойдет ничего, кроме очистки окна; но на самом деле теперь вы в другом терминале. Наберите **ls -l** для вывода содержимого текущего каталога, затем нажмите Ctrl+a с — и все. Перейдите в другой каталог и снова скомандуйте **ls**, нажмите Ctrl+a n, и вы увидите исходный список. При нажатии Ctrl+a с был создан новый терминал, а старый остался нетронутым; а при нажатии Ctrl+a n происходит

переключение между терминалами. Теперь нажмите Ctrl+a d для выхода из *Screen* и возврата в исходную оболочку, откройте еще один *xterm* и наберите **screen -r**.

Screen хорошо работает через SSH; можно подключиться с одного компьютера, выполнить команду, отключиться и снова подключиться к тому же сеансу *screen* через SSH с другого компьютера. Поэтому *screen* идеально подходит для системного администрирования.

например, обновлять пакеты или монтировать устройства — просто внимательно следите за тем, куда пойдет вывод команд.

Dsh — быстрая и простая команда, если вы не пытаетесь выполнять сложных или интерактивных команд на нескольких компьютерах (**dsh** с этим справится, но в выводе легко запутаться). Другая программа, которую мы рассмотрим, делает это иначе. *ClusterSSH* (хотя команда, которую вы наберете, называется просто **cssh**) тоже запускает команды на нескольких хостах через SSH, но другим способом. Эта программа, как и при обычном подключении через SSH, создает интерактивный сеанс, который выполняет одни и те же действия на нескольких компьютерах. Начнем с команды

```
cssh host1 host2 host3 ...
```

Как и в обычном SSH (или **dsh**), в списке хостов также можно указывать имена пользователей и номера портов. После этого *ClusterSSH* открывает отдельный *xterm* для каждого из хостов, где можно набрать команды для выполнения на этом компьютере. Пока ничего особенного не заметно — это мало чем отличается от открытия нескольких окон терминалов. Вся мощь **cssh** — в маленьком дополнительном окне, называемом административной консолью. Наберите в нем что-нибудь, и это появится во всех терминалах. Весь ввод с клавиатуры, включая сочетания клавиш, передается в терминалы, поэтому можно остановить все процессы, нажав Ctrl+C, или выйти из всех терминалов, нажав Ctrl+D в административной консоли. Единственное исключение — комбинации клавиш, используемые самой **cssh**. Как и с **dsh**, в командной строке можно указывать имена пользователей и номера портов:

```
cssh user1@host1 user2@host2:5555
```

Также можно задать группы хостов, называемые кластерами. Их проще всего задать в файле **/etc/clusters**, по кластеру на строку. Кластеры могут включать как хосты, так и другие кластеры:

```
servers root@host1 root@host2 ...
```

```
desktops host3 user@host4
```

```
all desktops servers
```

Специального синтаксиса для указания кластеров в **cssh** нет, они указываются так же, как имена хостов, и их легко спутать:

```
cssh desktops root@server1
```

Кластеры также можно задать в файле пользовательских настроек **~/clusterssh/config**. Он создается при первом запуске **cssh** этим пользователем, с закомментированными параметрами

по умолчанию. Здесь кластеры задаются немного иначе: сначала указываются имена кластеров, а потом — их содержимое.

```
clusters = desktops servers all
```

```
servers = root@host1 root@host2 ...
```

```
desktops = host3 user@host4
```

```
all = desktops servers
```

Добавление общих команд

В консоли администрирования есть различные меню. В меню Hosts [Хосты] можно добавить и удалить хосты и временно отключить отдельные хосты от консоли. Когда хост отключен, команды в терминале выполнять все еще можно, но то, что набирается в консоли, на него не отправляется. Снова выберите хост, чтобы подключить его.

Меню Send [Отправить] выглядит маловыразительно: в нем всего один пункт, который выводит имя каждого хоста в его терминал. Хотя иногда это бывает полезно, можно сделать гораздо больше, так как в меню допускается добавлять свои пункты. В разделе **send_menu_xml_file** файла **~/clusterssh/config** содержится ссылка на файл с меню, по умолчанию это **~/csshrc_send_menu** (да, имена конфигурационных файлов довольно бессистемны). Чтобы выполнить команду **uname -a** на каждом хосте, создайте файл и добавьте в него следующее:

```
<send_menu>
```

```
<menu title="System info">
```

```
<command>uname -r%n</command>
```

```
<accelerator>ALT-i</accelerator>
```

```
</menu>
```

```
</send_menu>
```

Символ **%n** отправляет символ «возврат каретки (CR)»; без него команда будет выведена в строке приглашения, но не выполнена. Другие шаблоны — **%s** для имени хоста, **%u** для имени пользователя и **%h** для имени хоста компьютера, на котором была запущена **cssh**. Параметр **accelerator** добавляет комбинацию клавиш.

В файле конфигурации по умолчанию есть много параметров, описанных на map-странице, но по большей части менять ничего не нужно. Все, что нужно сделать — задать несколько кластеров и, возможно, несколько пунктов меню Send, чтобы лишний раз не набирать команды. **LXF**

Drupal: Сайты за

Шашанк Шарма нашел способ быстро создавать сайты на *Drupal*. Правда ли, что *Drupal Distributions* — все, что ему нужно, или он просто легко возбудим?



Наш эксперт

Шашанк Шарма более четырех лет пишет о свободном ПО для различных изданий, включая Linux.com. Он соавтор книги «Fedora для начинающих».

В установочный пакет *Drupal* по умолчанию входит подборка модулей и тем. Но если они не отвечают вашим потребностям, придется пробовать тысячи модулей, чтобы найти те, что содержат нужные вам функции. В таком подходе нет ничего неправильного, но он требует времени. В зависимости от того, какой сайт вы хотите создать, на поиск, установку и настройку дополнительных модулей и их зависимостей может уйти неделя и больше. Хорошую альтернативу представляют *Drupal Distributions* [Сборки *Drupal*].

Наряду с ядром *Drupal* сборка включает все модули и настройки, необходимые для создания сайта. Преимущество сборки в том, что не нужно искать и настраивать все модули, необходимые для того или иного типа сайтов. Нужно лишь загрузить и установить сборку, точно так же, как и обычный релиз *Drupal*.

Если вы часто проделываете одно и то же в разных установках *Drupal*, сборки — то, что вам нужно. Например, если сначала вы всегда устанавливаете одни и те же пять модулей, можно создать сборку, в которой эти пять модулей установлены и активированы, а затем установить эту сборку вместо обычного релиза *Drupal*, чтобы сэкономить время.

Разбирая сборки

Сборка состоит из базового ядра *Drupal* и всех модулей, необходимых для конкретного типа сайта. Возьмем, например, сборку *Kickstart*. Наряду с ядром *Drupal* и стандартным набором модулей и тем эта сборка включает дополнительные модули и темы, в помощь созданию собственного коммерческого сайта. После установки мы получаем рабочий, готовый к использованию коммерческий сайт, на котором есть корзина и все прочее, подобающее такому сайту.

Самая важная часть сборки — профиль установки. Его можно считать фундаментом сборки. Если вы раньше устанавливали *Drupal 7*, то, возможно, уже замечали два профиля установки. При установке *Drupal* мастер просит выбрать между профилями установки: *Standard* [стандартный] и *Minimal* [минимальный].

Профили установки содержат описание модулей и темы, которые нужно сделать доступным и активировать после установки *Drupal*. Профиль установки сообщает *Drupal*, какие модули, настройки и другие настройки должны использоваться с этой установкой *Drupal*. Важно понять разницу между профилями установки и сборками.

Профиль установки не связан ни с какими модулями и темами. Это только список всех настроек, которые вы хотите использовать в своей установке. Например, если вам захочется установить профиль установки для блоггинга, он будет содержать настройки (тип контента, роли, права доступа и т.д.), которые позволяют нескольким пользователям создавать контент для сайта. Но если профилю установки нужны дополнительные модули, например, модуль оценки контента сайта, эти модули вам придется устанавливать вручную.

С другой стороны, сборка включает базовое ядро *Drupal* и дополнительные модули и темы. По существу, именно благодаря простоте, с которой можно получить готовый сайт, не прилагая никаких усилий, многие новостные сайты, правительственные и некоммерческие организации выбирают сборки *Drupal* для создания своих сайтов.

«Важно понять разницу между профилями установки и сборками.»

Разные сборки

Хотя две или несколько сборок могут быть рассчитаны на одну и ту же аудиторию, бешеное количество разных модулей *Drupal* гарантирует, что двух похожих модулей среди них нет.

Сборка довольно сильно отличается от стандартного релиза *Drupal*. И не только функциями дополнительных модулей, добавленных в сборку.

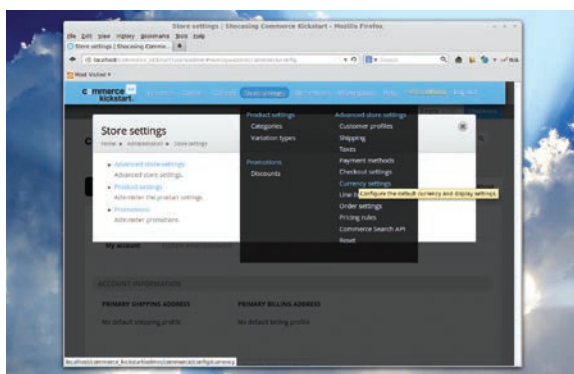
В сборке почти всегда нестандартное меню администратора. Для *Drupal* есть несколько различных модулей навигации, и создатели сборок обычно выбирают один из этих альтернативных модулей вместо традиционного меню в стандартных релизах *Drupal*. Выбор модуля навигации зависит от целевого назначения сборки.

На самом деле, в некоторых сборках, например в *Commerce Kickstart*, есть собственное административное меню. Для таких сложных и богатых на возможности сборок стандартное меню навигации и администрирования непрактично.

Независимо от того, как выглядит сборка, ее административные функции те же самые. Это означает, что запуск заданий *Cron*, применение обновлений, управление модулями и другие административные задачи, которые можно выполнять в *Drupal*, решаются точно так же, как и в обычном релизе *Drupal*.

Теперь, когда мы рассмотрели основы сборок, пора создать собственную сборку специального назначения.

Мы уже сказали, что самой важной частью сборки является профиль установки. Его можно считать фундаментом сборки. Чтобы создать сборку, сначала нужно создать для нее профиль установки.



» Меню навигации не кажется таким уж важным, но может решить судьбу сборки.

НОЛЬ СЕКУНД

Управление Drupal с Drush

В *Drush*, командной оболочке *Drupal*, легко сделать внутреннюю очистку, провести свежую установку *Drupal*, установить и включить модули, создать пользователей и выполнить разные другие действия, не притрагиваясь к мыши.

Хотя *Drush* есть в репозиториях большинства дистрибутивов Linux, лучше всего установить его через официальный канал PEAR. Если у вас не установлен PEAR, можете воспользоваться менеджером пакетов своего дистрибутива. Команда `sudo apt-get install php-pear` делает всю работу.

Также нужно установить пакеты *PHP5-mysql* и *PHP5-gd*. Теперь все готово для установки *Drush*:

```
pear channel-discover pear.drush.org
pear install drush/drush
```

Установив *Drush*, легко установить *Drupal* и соответствующие модули из командной строки. Убедитесь, что у пользователя есть права доступа на запись в каталог, и запустите команду

```
drush dl drupal
```

Она загрузит последний релиз *Drupal* с сайта проекта в текущий каталог. Зайдите в созданный каталог `drupal-7.23`.

Теперь можно установить *Drupal* командой `drush si standard --db-url=mysql://dbusername:dbpassword@localhost/dbname - site-name="Drupal site with Drush"`

Эта команда аналогична установке с помощью мастера в браузере. Сначала мы указываем профиль установки, которым хотим воспользоваться (в данном случае, стандартный), затем указываем параметры базы данных, созданной для нашего сайта.

Ну вот, *Drupal* установлен, и можно установить модули на новый сайт командой `drush dl modulename`

Первое, что нужно сделать — создать сайт на *Drupal*, который вы хотите превратить в сборку. Убедитесь, что вы установили все желаемые модули и темы. Контент не добавляйте, потому что из этого установленного сайта мы будем делать сборку.

Профиль установки содержит три существенных файла: **profilename.info**, **profilename.install** и **profilename.profile**. Содержимое этих файлов определяется тем, какие модули и функции должны содержать сборка.

Файл **profilename.info** в минимальном случае содержит имя профиля, краткое описание профиля, ядро *Drupal*, для которого собран профиль, и список модулей, которые нужно включить автоматически.

```
name = Profile name goes here
description = Description of what the profile does.
core = 7.x
dependencies[] = views
dependencies[] = ctools
dependencies[] = advanced_menu
```

В стандартный релиз *Drupal* входят файлы **.info** для стандартного и минимального профилей установки. Они находятся в каталоге **profiles/**.

Файл **.install** — PHP-скрипт с командами, которые добавляют содержимое в базу данных во время установки. Ниже приведен файл **.install** для профиля установки Minimal:

```
<?php
/**
 * Implement hook_install().
 *
 * Perform actions to set up the site for this profile.
 */
function profilename_install() {
  include_once DRUPAL_ROOT . '/profiles/minimal/minimal.install';
  minimal_install();
}
```

?>

Наконец, у нас есть файл **.profile**. Это тоже PHP-файл, но он используется, когда нужно добавить этапы в процесс установки. Так, если вы хотите, чтобы пользователи указали текущее географическое положение или поддерживаемую ими политическую партию, это нужно добавить в файл **.profile**.

Были времена, когда приходилось тратить кучу времени, вплоть до часов, на написание этих файлов вручную, даже если вы хорошо знали внутренности *Drupal* и PHP. Но эти времена прошли. Модули **Profiler** и **Profiler Builder**, в соединении с *Drush*, могут создать эти файлы за несколько минут.

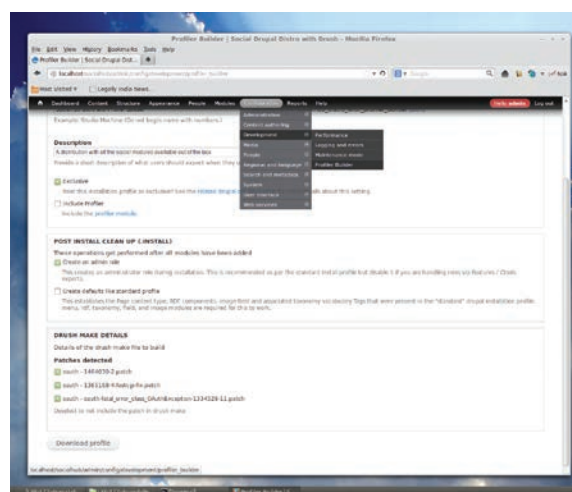
Запустив сайт, можно загрузить модули командами

```
drush dl profiler
```

и

```
drush dl profiler_builder
```

»



» С помощью модуля **Profiler Builder** можно волшебным образом создать профили установки.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

а затем включить их командами

```
drush en profiler
```

и

```
drush en profiler_builder
```

Наш сайт под названием Socialhub расположен в каталоге `/opt/lampp/htdocs/socialhub`. Откройте терминал и перейдите в этот каталог. Затем выполните команду

```
drush distro socialhub
```

чтобы создать профиль установки. Убедитесь, что у вас есть права доступа на запись в этот каталог, или при необходимости воспользуйтесь `sudo` или `su`.

```
$ cd /opt/lampp/htdocs/socialhub
```

```
$ drush distro socialhub
```

```
Wrote .tar file socialhub.tar to current directory [ok]
```

```
$ tar tvf socialhub.tar
```

```
-rw-r--r-- 501/501 1360 2013-09-25 14:12 socialhub/socialhub.
```

```
info
```

```
-rw-r--r-- 501/501 3609 2013-09-25 14:12 socialhub/socialhub.
```

```
install
```

```
-rw-r--r-- 501/501 709 2013-09-25 14:12 socialhub/socialhub.
```

```
profile
```

```
-rw-r--r-- 501/501 1623 2013-09-25 14:12 socialhub/drupal-org.
```

```
make
```

```
-rw-r--r-- 501/501 498 2013-09-25 14:12 socialhub/local.make.
```

```
example
```

Команда `drush distro` создает файлы профиля установки и упаковывает их в TAR-файл. Для просмотра его содержимого используйте команду `tar tvf`. Как видите, команда генерирует файлы `.install`, `.profile` и `.info`, а также файл `.make`, который понадобится для сборки. В качестве альтернативы можно воспользоваться командой

```
drush distro socialhub --untar
```

Она создаст в текущем каталоге подкаталог `socialhub`, содержащий файлы профиля установки.

Если вам неудобно в командной строке, профиль установки можно создать и в графическом интерфейсе. Зайдите на свой сайт *Drupal* как администратор и выберите Configuration > Development > Profiler Builder [Настройка > Разработка > Сборщик профилей].

Укажите всю информацию, такую как имя и описание профиля, и когда закончите, нажмите кнопку Download profile [Загрузить профиль] в нижней части страницы. Как и команда

```
drush distro/
```

модуль Profiler Builder создаст файл TAR, содержащий файлы `.info`, `.profile` и другие файлы.

Хотя сборщик профилей идеально подходит для создания профиля установки, с его помощью нельзя создать сборку с заданными настройками. Этот модуль создаст лишь установочный профиль, который отражает все модули и темы, использованные при создании вашего сайта. Он не содержит информации о настройках. Все дополнительные модули для вашей сборки находятся в каталоге `sites/all/modules/contrib`. При установке *Drupal* из своей сборки у вас будут все модули и темы, но их все равно придется настраивать.

Если вы хотите сохранить и настройки модулей, нужно воспользоваться модулями Features [Возможности]. С их помощью фактически можно экспортировать настройки и конфигурацию вашего сайта *Drupal*.

Сборка сборок

У вас уже есть файл TAR, содержащий профиль установки вместе с файлом `drupal-org.make`. Этот тот файл `.make`, который

используется для создания сборки. Извлеките файлы из TAR-файла. Файл `.make` должен выглядеть примерно так:

```
; social_drupal_distro_with_drush make file for d.o. Usage core = "7.x" api = "2"; +++++ Modules +++++ projects[admin_menu] [version] = "3.0-rc4"
```

```
projects[admin_menu][subdir] = "contrib"
```

```
projects[ctools][version] = "1.3"
```

```
projects[ctools][subdir] = "contrib"
```

```
projects[profiler_builder][version] = "1.0"
```

```
projects[profiler_builder][subdir] = "contrib"
```

```
projects[simplenews][version] = "1.0"
```

```
projects[simplenews][subdir] = "contrib"
```

```
projects[media][version] = "1.3"
```

```
projects[media][subdir] = "contrib" ///END CODE///
```

Это лишь фрагмент нашего файла `make`. Этот файл нужно от-крыть и добавить строку

```
projects[] = drupal
```

на новой строке под `api = "2"`, чтобы получилось следующее:

```
; social_drupal_distro_with_drush make file for d.o. usage
```

```
core = "7.x"
```

```
api = "2"
```

```
projects[] = drupal
```

Все прочее в файле останется прежним. Так как мы собираем свою сборку с помощью *Drush*, эта строка сообщает *Drush*, что *Drupal* — основной компонент нашей сборки. Теперь воспользуемся файлом `.make` для генерации сборки командой

```
drush make --prepare-install --tar socialhub/drupal-org.make socialhub
```

Этот процесс может длиться некоторое время, зависящее от объема сайта, который определяется количеством модулей и тем.

При выполнении этой команды *Drush* загрузит стандартный релиз *Drupal* вместе со всеми дополнительными модулями и темами, использованными вами для создания своего сайта, и создаст файл `socialhub.tar.gz` в текущем каталоге.

Для проверки сборки распакуйте файл `.tar.gz`. Затем создайте новую базу данных и пользователя. Наконец, укажите в браузере каталог, куда распаковали сборку, чтобы начать установку. Своей сборкой даже можно поделиться со всем миром. Но если вы хотите разместить ее на сайте *Drupal*, это делается иначе. Вместо всего пакета, создаваемого командой

```
drush make --prepare-install
```

для размещения сборки на Drupal.org понадобится только файл `.make`. Это сделает команда

```
drush make --generate-makefile drupal-org.make
```

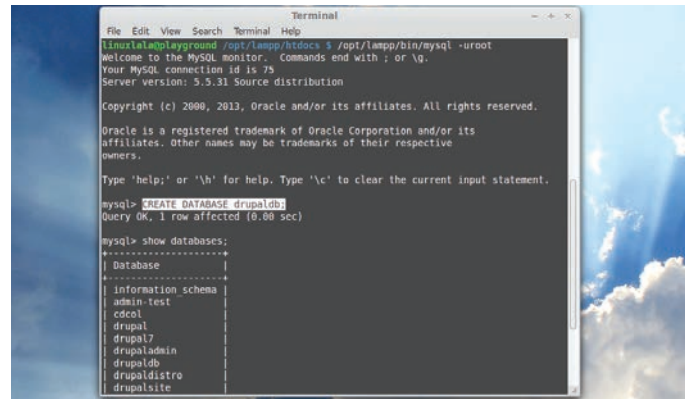
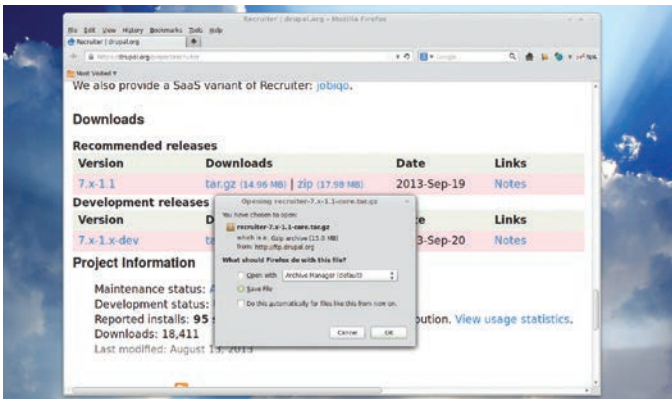
Подробная информация о размещении сборки на Drupal.org приведена на <https://drupal.org/node/642116>.

Пользовательские модули

Сборка по большей части — лишь способ получения установки *Drupal* с более богатой функциональностью, предоставляемой встроенными модулями. К модулям, как и в обычной установке *Drupal*, можно применять обновления. На самом деле, в большинстве популярных сборок предоставляются регулярные обновления для ядра *Drupal* и дополнительных модулей, включенных в сборку.

Но есть некоторые сборки, построенные вокруг пользовательских модулей, созданных самими разработчиками. При использовании таких сборок не остается ничего другого, кроме как ждать, пока разработчик не выпустит обновления для модуля и сборки. Недостаток специальных сборок в том, что вы можете угодить в ситуацию, когда сайт будет сложно обновить, если разработчик прекратит поддержку сборки.

Устанавливаем сборку

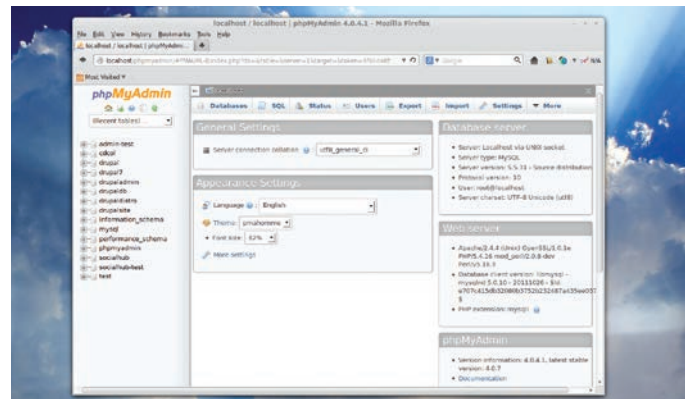
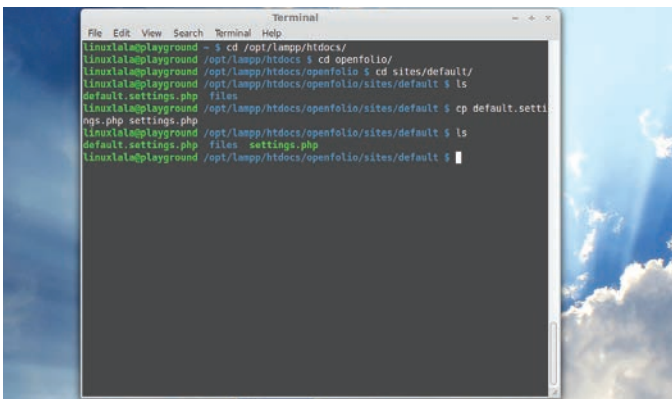


1 Загрузим сборку

Загрузите tar-архив сборки с сайта проекта и распакуйте его в соответствующий каталог. В зависимости от настроек системы это может быть каталог в `htdocs` или в `/var/www/`. При желании переименуйте созданный при распаковке каталог.

2 Создадим базу данных и пользователя

Для создания базы данных и пользователя можно воспользоваться командной строкой *MySQL* или *PHPMyAdmin*. Держите имя пользователя и базы данных под рукой, так как *Drupal* попросит указать их во время установки. Делать это придется даже при установке стандартного релиза *Drupal*.

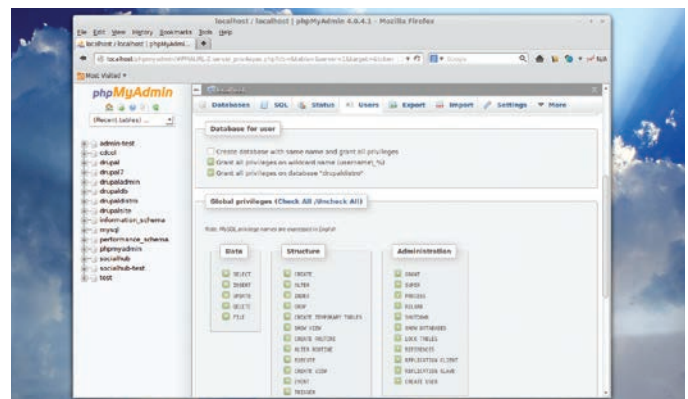
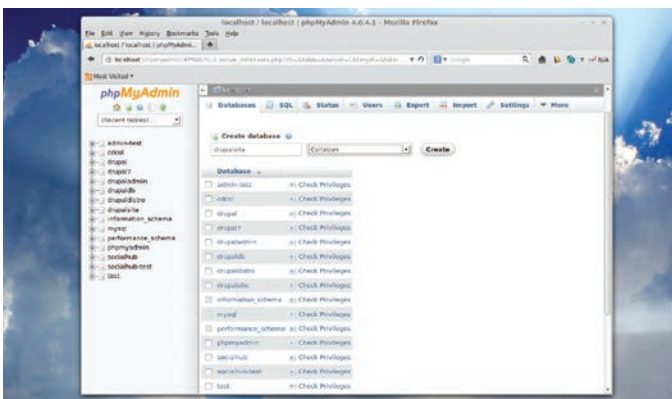


3 Переименуем default.settings.php

Зайдите в подкаталог `sites/default` в распакованном каталоге и переименуйте файл `default.settings.php` в `settings.php`. Откройте распакованный каталог в браузере и продолжите установку в браузере.

4 Доступ через PHPMyAdmin

Зайдите в *PHPMyAdmin* как администратор. Кроме создания пользователей и баз данных, в *PHPMyAdmin* также можно изменять базы данных и выполнять SQL-запросы. Нажмите кнопку *Databases* [Базы данных] в левом верхнем углу основного окна.



5 Создаем базу данных

Укажите имя базы данных и нажмите кнопку *Create* [Создать]. На этой странице будут указаны существующие базы данных. Среди них вы найдете созданную базу данных.

6 Создаем пользователя

Нажмите ссылку *Check Privileges* [Проверить привилегии] рядом с созданной базой данных. В разделе *New* [Создать] нажмите *Add User* [Добавить пользователя]. Укажите имя пользователя и пароль и назначьте ему привилегии. Нажмите *Go* [Вперед] в нижней части страницы. **LXF**

Zabbix: Следим

Zabbix — это приложение сетевого мониторинга, которое отслеживает все. Шашанк Шарма даже подумывает, не оживший ли это «глаз Саурана».



Наш эксперт

Шашанк Шарма пишет о свободном ПО более четырех лет, в том числе и для Linux Foundation. Он — соавтор *Beginning Fedora*.



Скрытый в своей крепости, системный администратор видит все. Его око проникает в дистрибутивы, устройства, диски и приложения. Ты же понимаешь, о чем я говорю, мой верный читатель: то самое великое око, лишенное век, обрамленное пламенем. Ну, насчет век и пламени — отставить, ведь речь тут не о вымысле!

Zabbix — приложение сетевого мониторинга, позволяющее отследить все, что происходит в сети. Независимо от размера сети и операционных систем, установленных на составляющих ее машинах, Zabbix умеет контролировать каждый аспект этих машин. С помощью Zabbix можно отслеживать свободное место на диске, загрузку CPU, статистику сетевого трафика, потребление памяти,

статус приложений, контрольную сумму файла и многое другое. Вся собранная информация хранится в базах данных, и вы можете употребить MySQL, Oracle или PostgreSQL, поскольку Zabbix поддерживает их все.

Конкурентное преимущество

Однако с чего вам хвататься за Zabbix, когда большинство вокруг вас постоянно бросаются такими названиями, как Cacti и Nagios?

Zabbix, как и его более популярные собратья, поддерживает Simple Network Management Protocol (SNMP). А значит, эти приложения могут осуществлять мониторинг почти любого устройства в сети, например, роутеров, принтеров, переключателей, модемов и т. д.

Nagios, хоть и является промышленным стандартом для мониторинга локальных и удаленных машин и инфраструктуры ИТ, почти всегда применяется совместно с другими инструментами мониторинга. Дело в том, что у Nagios нет функции визуализации, и он не способен превратить записанные данные в легко понимаемые графики и схемы. Но все же у Nagios имеется превосходная функция оповещения обо всех устройствах, которые он отслеживает.

У Cacti, напротив, нет механизма оповещения, зато он умеет создавать прекрасные и подробные графики, которые представляют все данные, собранные им об устройствах в сети.

А вот выбранный нами инструмент очень удобен и для оповещений об опасности, и для создания графиков. Мало того, что он рассылает оповещения об угрозах по электронной почте, через SMS и Jabber — он позволяет реагировать автоматически. То есть Zabbix можно настроить так, чтобы он предпринимал заданные действия на удаленных машинах, если произошли определенные события.

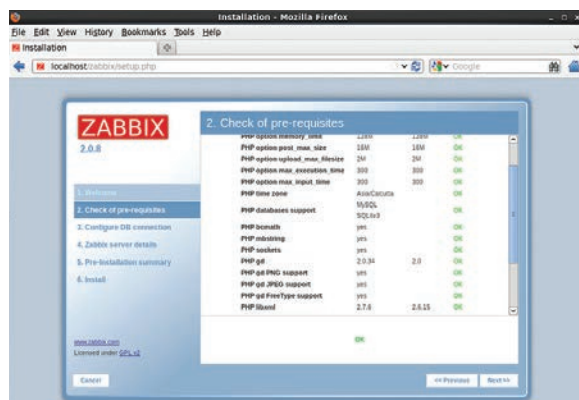
Говоря о визуализации, Zabbix поддерживает создание в реальном времени графиков отслеживаемых проблем, а заодно предлагает карты сети, изображения и даже слайд-шоу, что очень удобно для сложных сетей, чьи администраторы предпочли бы видеть графическое представление данных.

Осваиваем Zabbix

Приложение мониторинга в первую очередь выполняет две функции. Оно собирает данные по всем устройствам сети и передает их центральному управляющему органу. Zabbix использует два ключевых компонента для выполнения этих функций — zabbix-server и zabbix-agent.

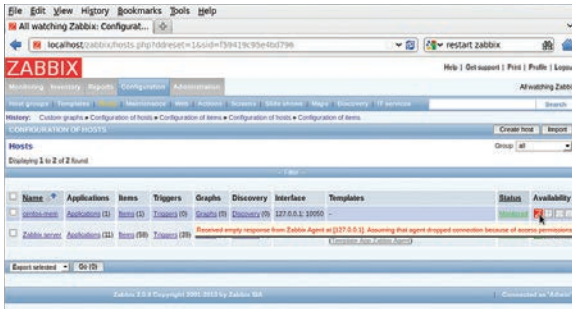
Как и предполагает его название, zabbix-agent устанавливается на все машины, мониторинг которых вы хотите осуществлять в сети. Агент проводит мониторинг ресурсов оборудования, а также приложений, и передает всю информацию на zabbix-server. На данный момент zabbix-agent доступен для Linux, FreeBSD, OpenBSD, NetBSD, HP-UX, MacOS X, Solaris и разных версий Windows.

Другая часть, zabbix-server, отвечает за сбор и анализ данных. В случае появления проблем или простоя сервер отправляет уведомление на настроенные учетные записи. На всю сеть достаточно установки одного экземпляра zabbix-server.



Все зеленые ОК подтверждают: я знаю, что я здесь делаю.

За своей сетью



➤ Красный означает «дальше хода нет», но, к счастью, при наведении мыши *Zabbix* даст пояснения.

С *Zabbix* вы получаете централизованный web-интерфейс, через который можно управлять тем, какие события подвергать мониторингу в своей сети. Из-за изобилия функций и аспектов, которые он надеется охватить, многим пользователям интерфейс может иногда казаться сложным и громоздким. Но несмотря на ваши сомнения и первичные впечатления, мы рекомендуем вам пуститься в путешествие по *Zabbix*.

Устанавливаем Zabbix

Проект предлагает бинарники для нескольких дистрибутивов, таких, как Debian и CentOS, и в репозиториях многих дистрибутивов *Zabbix* есть. Однако, как это часто бывает с инструментами администрирования, репозитории не всегда предлагают самую последнюю версию.

В качестве базы, на которой мы установим *zabbix-server*, мы взяли CentOS. В сети есть другие машины с другими версиями Linux: например, Linux Mint.

У вас уже должны быть установлены нужные пакеты PHP и MySQL, но если вы устанавливаете *Zabbix* на свежую систему, следующая команда обеспечит зависимости:

```
su -c "yum -y install mysql-server php5-mysql php5-gd"
```

Лучший способ установить *Zabbix* на CentOS — через репозиторий EPEL, поскольку основные репозитории не предлагают самых свежих версий. Если на вашей машине он пока не установлен, сначала придется настроить репозиторий EPEL:

```
su -c "yum -y localinstall http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm"
```

```
su -c "yum -y install zabbix20-server-mysql zabbix20-webmysql"
```

Первая команда настроит и запустит репозиторий. Во второй мы указываем, какие пакеты хотим установить — в данном случае это *Zabbix 2.0.x*.

Zabbix поддерживает базы данных MySQL, PostgreSQL и Oracle, и может хранить собранные данные в любой из них. Настройка по умолчанию рассчитывает на наличие базы данных под названием *zabbix*. Имя пользователя по умолчанию для этой базы данных — *zabbix*, и пароль не требуется.

Если вы захотите изменить какие-либо из этих предустановленных настроек, можно отредактировать файл `/etc/zabbix/zabbix_server.conf`.

Если что-то не ладится

Проект предоставляет отличную документацию, но если вы никогда не работали с приложениями сетевого мониторинга типа *Zabbix*, что-то может пойти не так. Поэтому вам не стоит устанавливать его на рабочий сервер с бухты-барухты.

Неверно настроенный сетевой монитор типа *Zabbix* способен истощить ваши ресурсы. База данных съест немалую долю дискового пространства, если вы не настроите ее на удаление старых данных. У вас также могут быть

причины избегать *Zabbix*. Например, вы можете решить, что *Zabbix* не оптимален для ваших целей, поскольку это будет перебором.

Сама документация не дает инструкции по удалению *Zabbix*, так что же вам делать? Естественно, читать дальше!

Самое разумное решение — отключить *Zabbix*. Но сначала перейдите в `Configuration > Hosts` и отключите настроенные хосты. Далее, удалите *zabbix-agent* со всех машин в сети и остановите *zabbix-server*.

Решив изменить настройки базы данных, позаботьтесь, чтобы они были вам удобны, так как они очень скоро нам понадобятся, когда мы будем создавать базу данных для *Zabbix*. Вам нужно будет запустить из терминала следующие команды:

```
mysql -uroot -e 'create database zabbix'
```

```
mysql -uroot -e "create user 'zabbix'@'localhost' identified by 'ChangeMe'"
```

```
mysql -uroot -e "grant all on zabbix.* to 'zabbix'@'localhost'"
```

```
mysql -uroot -e "flush privileges"
```

Приведенные команды очевидны. Мы создаем базу данных в первой команде, и пользователя во второй. Убедитесь, что вы указали верное имя пользователя и тот пароль, который записан в файле `/etc/zabbix/zabbix_server.conf`. Не мешает изучить тонкости администрирования MySQL.

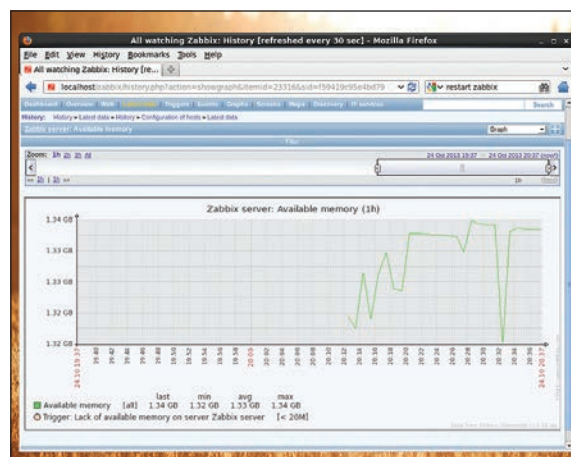
Теперь, когда база данных на месте, мы готовы ее заселить:

```
mysql -uzabbix -p'ChangeMe' zabbix < /usr/share/zabbixmysql/schema.sql
```

```
mysql -uzabbix -p'ChangeMe' zabbix < /usr/share/zabbixmysql/images.sql
```

Скорая помощь

Поскольку все данные записаны в базе данных, чем больше у вас ОЗУ, тем быстрее будет к ним доступ.



➤ Можете использовать панель сверху справа от графика, чтобы указать временной период.

```
mysql -uzabbix -p'ChangeMe' zabbix < /usr/share/zabbixmysql/
data.sql
```

Серверный компонент *Zabbix* содержит три отдельных части: *zabbix-server*, базу данных и графический интерфейс. Два первых компонента мы установили; осталось установить интерфейс на основе браузера.

Наладим интерфейс

Прежде чем приступить к установке интерфейса, следует убедиться, что *zabbix-server* работает. Если нет, его запустит команда **zabbix-server**. Теперь вы можете нацелить браузер на адрес <http://localhost/zabbix>, где находится мастер, который и проведет вас по всем стадиям установки.

Щелкните Next внизу справа, чтобы начать установку. *Zabbix* будет сообщать обо всех ошибках, который ему встретятся в файле настройки PHP **/etc/php.ini**. Ошибки будут выделены красным, и мастер сообщит вам о текущих значениях, указанных в файле для ошибочных переменных, и о значениях, ожидаемых *Zabbix*.

Установка не начнется, пока вы не исправите все обнаруженные ошибки. К сожалению, вам придется вручную редактировать файл, чтобы внести рекомендуемые изменения. На новой установке PHP нужно будет изменить часовой пояс, значения `post_max_size`, `max_execution_size` и `max_input_time`.

Внеся все изменения, сохраните файл **/etc/php.ini** и затем запустите

```
apachectl restart
```

что обеспечит использование новых настроек.

Вернитесь в мастер и щелкните по Refresh внизу экрана. Если вы чародей и обладаете властью преобразовывать файлы настройки по своему желанию, *Zabbix* позволит вам продолжить установку.

Щелкните Next и укажите информацию по базе данных, созданной ранее, а именно: имя базы данных, имя пользователя и пароль.

Следующие несколько шагов в мастере довольно обычные. Вам предьявят указанную вами информацию для проверки, и, наконец, интерфейс будет установлен. После щелчка на Finish вас автоматически перенаправят на экран приглашения.

Так завершается серверная сторона настройки *Zabbix*. Теперь вы можете установить *zabbix-agent* на всех машинах в сети, мониторинг которой намерены осуществлять. В CentOS этот фокус erlediet команда

```
su -c "yum -y install zabbix20-agent"
```

Имя пользователя по умолчанию для вновь установленного интерфейса администратора — “admin”, а пароль — “zabbix”.

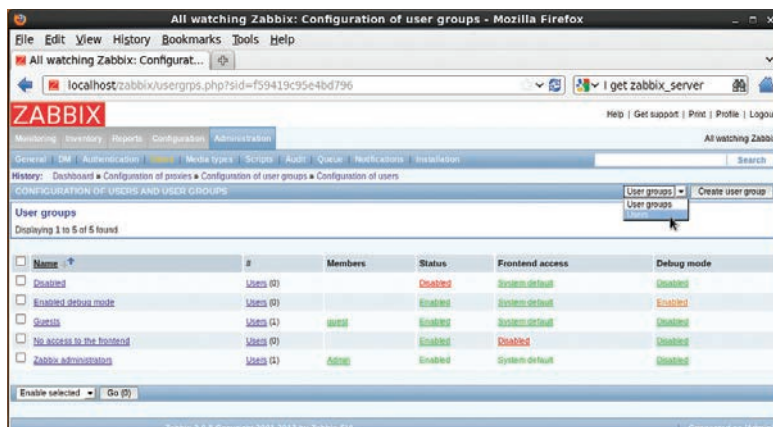
Впервые войдя в *Zabbix*, вы, возможно, слегка растеряетесь, из-за обрушивающейся лавины информации. Но не волнуйтесь, скоро мы вас познакомим, и затем вы сами освоите *Zabbix* и все, что он предлагает.

Скорая помощь



Базы данных — узкое место *Zabbix*. Обеспечьте правильные настройки своей базы данных, чтобы не снижать производительности всей системы.

> Zabbix не любит пользователей-одиночек: пользователь должен принадлежать по крайней мере к одной из групп.



Однако перед тем, как продолжать, вам придется поменять пароль. Щелкните Profile вверху справа, после чего щелкните по кнопке Change Password. Введите новый пароль и щелкните Save внизу.

Поскольку *Zabbix* — огромный проект с огромным же количеством функций, их очень трудно собрать вместе. Вся настройка распределяется между пятью основными вкладками верхней панели меню: Monitoring, Inventory, Reports, Configuration и Administration.

При наведении мыши на любую из этих кнопок второе меню отразит, что именно кнопка делает. Например, если вы наведете мышь на Monitoring, второе меню покажет Overview, Web, Graphs, Triggers и т.д.

По сути, второе меню — это подменю, которое показывает опции выбранной основной вкладки.

Отслеживайте все

Хотя самым проверенным методом для освоения чего бы то ни было является метод проб и ошибок, к *Zabbix* это не относится. Можно, конечно, попытаться разобраться со всеми вкладками, меню, подменю, выпадающими списками и прочим, но у вас все получится гораздо лучше, если вы не пожалеете времени на изучение руководства по работе с *Zabbix* (<https://www.zabbix.com/documentation/2.0/manual>).

Каждая часть сети, которую вы хотите подвергнуть мониторингу, будь то принтер, коммутатор, файл, роутер, физический сервер или виртуальная машина и т.д., определяется как хост. Чтобы начать мониторинг чего-либо, сначала нужно создать хост. Для добавления хоста перейдите в Configuration > Hosts и щелкните по кнопке Create host справа. Добавив хост, надо выбрать как минимум одну группу хостов, потому что разрешения привязываются не к отдельным хостам, а к их группам. Введите остальную информацию, например, IP-адрес хоста, а закончив, щелкните Save. Более подробную информацию вы найдете в <https://www.zabbix.com/documentation/2.0/manual/config/hosts>.

Теперь ваш вновь созданный хост отобразит список настроенных хостов. Щелкните по ссылке Items [Пункты] своего хоста.

Пункты вы должны определить для каждого созданного вами хоста. Они указывают, какие именно данные надо собирать для каждого из настроенных хостов. Если вы хотите сделать мониторинг нагрузки на CPU, проверить использование приложением памяти, статистику диска и т.п., вы должны описать позиции для каждой задачи. Самый важный элемент при определении пунктов — код пункта. Именно по нему определяется то, что вы хотите подвергнуть мониторингу *Zabbix*. Щелкните по кнопке Select рядом с полем Key и выберите код из списка. Например, если вам нужен мониторинг использования CPU, выберите system.cpu.load key.

Вернемся в Configuration > Hosts. Если все пойдет хорошо, под заголовком Availability вы увидите символ зеленого сигнала. Это означает, что и хост, и пункт настроены должным образом. Увидев же красный символ, наведите на него мышь, чтобы узнать, в чем проблема. В зависимости от ошибки вам придется изменить настройку пункта или хоста. Более подробную информацию ищите на <https://www.zabbix.com/documentation/2.0/manual/config/items/item>.

Установка *Zabbix* предоставляет хост сервера *Zabbix* с 58 уже настроенными пунктами. Но по умолчанию этот хост не включен. Вы можете включить его и изучить в нем пункты, чтобы лучше понять, как работает *Zabbix* и что означает настройка хостов и пунктов.

Создав хосты и пункты, можете сразу переходить в Monitoring > Latest data, чтобы посмотреть, какую информацию *Zabbix* собрал о ваших настроенных хостах.

Если вы используете хосты сервера *Zabbix* по умолчанию, вы также можете увидеть собранные данные, представленные

в графиках. Под заголовком Local data выберите пункт, по которому хотите получить больше информации, и затем щелкните по ссылке графика с правого края.

Trigger наготове

Вам осталось еще создать Trigger [Пусковой механизм] для ваших настроенных хостов. Триггер используется для анализа собранных данных и сравнения их с приемлемыми значениями. В случае расхождения этих двух значений Zabbix выстрелит сообщением о проблеме. Определяя триггер, можно применять операторы, такие, как more than [больше, чем], less than [меньше, чем], not equal to [не равно], и т. п. Как и пункты, триггеры создаются из самого экрана Configuration > Hosts.

Когда вы это сделаете, финальным шагом будет настройка системы на отправку уведомлений — Notifications — в случае, если Zabbix обнаружит проблемы с собранными данными. Это делается в два этапа.

Сначала надо определить тип медиа, то есть механизм доставки, служащий для рассылки уведомлений, например, SMS, Jabber, электронная почта и т. п. Можете определить тип медиа со страницы Administration > Media types.

Далее определите действие в Configuration > Actions. Для каждого действия надо указать источник события, например, триггер, и операцию. Операция может быть или отправкой сообщения через указанный тип медиа, или выполнением удаленной команды. Эта последняя опция позволяет при выполнении определенных условий автоматически запустить предопределенную команду на удаленной машине. Вы можете использовать эту функцию для запуска критических команд: например, удаления более старых файлов и очистки /tmp на удаленной машине, если на ней исчерпается дисковое пространство, и т. д.

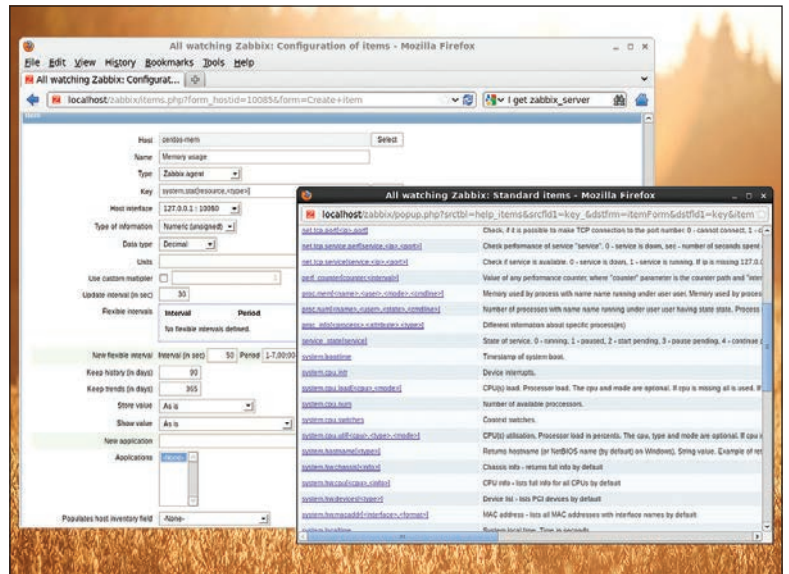
Если вы планируете отслеживать один и тот же набор параметров на всех машинах вашей сети, нет особого смысла несколько раз повторять все эти действия. Zabbix предлагает куда более практичное решение в виде шаблонов. Вы можете считать шаблоны набором предустановленных единиц, таких, как пункты, триггеры, действия и т. д. Применяя такой шаблон к каждому вновь создаваемому хосту, вы сэкономите время.

Вы можете настроить несколько учетных записей пользователя и групп, и воспользоваться всесторонней системой разрешений Zabbix для ограничения доступа к функциям администратора или подвергающимся мониторингу хостам.

Пользователи делятся на два класса: те, у кого есть доступ к управлению функциями интерфейса Zabbix, и те, у кого есть доступ только к подвергающимся мониторингу хостам в группах хостов.



➤ Перейдите в Configuration > Screens, если вы хотите сгруппировать информацию из разных источников и отобразить ее на одном экране.



➤ Ваш выбор пункта-Item, будь то агент Zabbix, SNMP и т. д., решит, какие параметры вы сможете отслеживать.

Пользователя можно классифицировать как User, Admin или Super Admin. Только Super Admin имеет полный контроль над любыми аспектами системы. Для других пользователей можно ограничить количество функций и устройств, мониторинг которых разрешается осуществлять. Например, Zabbix Admin имеет доступ к меню Monitoring и Configuration, но не имеет доступа к настройкам групп хостов. Однако вы можете добавить разрешения, предоставив такой доступ.

Чтобы создать нового пользователя, щелкните по вкладке Administration в первой панели меню, затем щелкните по Users во второй панели меню. Этот экран перечислит все настроенные группы пользователей. Щелкните по выпадающему списку User groups вверху справа на экране и выберите Users. Теперь щелкните по кнопке Create user.

Добавок к своему сложному в навигации интерфейсу, Zabbix не слишком интуитивен. Например, разрешения можно присваивать только группам пользователей, но не индивидуальным пользователям. Так что сначала вам придется создать группу пользователей, добавить вашего пользователя к новой группе, а затем определить разрешения для этой группы, чтобы у пользователя была большая степень доступа.

Шапочное знакомство

Должно быть, создалось впечатление, что мы слишком поспешно промчались по самой основной части руководства: инструкциям по использованию приложения.

К сожалению, с учетом обширности проекта, невозможно было углубиться в подробности. Но у проекта отличная документация, где подробно обсуждается каждый шаг. Более того, у него очень активное сообщество пользователей, к которому всегда можно обратиться в случае появления какой-то проблемы.

Zabbix — один из тех инструментов, на которые надо потратить немало времени, чтобы увидеть результат. Тем, кто хорошо разбирается в работе сети и знает, какие параметры хочет подвергнуть мониторингу, Zabbix может показаться довольно простым в использовании, даже если вы никогда раньше не использовали подобные приложения.

Львиная доля критики Zabbix касается его web-интерфейса, а уж никак не функций, которые он предоставляет. Хотя интерфейс иногда кажется неинтуитивным, разработчики заслуживают высоких похвал за предоставление отличного механизма для управления каждым аспектом Zabbix. LXF

Git: Как все это

Как Git ухитрится хранить ваш драгоценный код в безопасности? Джон Лэйн срывает покров тайны с другого проекта Линуса.



Наш эксперт

Джон Лэйн

Однажды зайдя в Linux в 1994 году, Джон так там и застрял. Он и не упомнит, когда в последний раз использовал Windows.



Поговорим о Git. Если вы когда-нибудь пытались установить программный пакет Linux, которого нет в стандартном репозитории вашего дистрибутива, есть немало шансов, что вам пришлось компилировать пакет из исходника. Вероятно, вам пришлось скачать этот самый пакет из системы контроля версий (VCS), и еще более вероятно, что VCS, которую вы использовали, называется Git.

Git — это VCS, которую Линус Торвальдс создал для исходников ядра Linux после того, как его пути с BitKeeper разошлись. Возможно, вы читали наше Сравнение в LXF157 (стр. 32), которое знакомо с хостингом и объясняло ее использование, или вы даже использовали ее для каких-то своих проектов. Однако в чем волшебство — как Git умудряется делать то, что он делает?

Познакомившись с данным руководством, вы узнаете, как Git хранит ваши файлы и их историю. Это позволит вам лучше понять ее концепцию, терминологию и инструменты. Если вы раньше не использовали Git, стоит обратить внимание на наши предыдущие руководства в архивах.

Вероятнее всего, в вашей системе Git уже есть, и большая часть того, что мы будем делать, сработает в любой оболочке (мы используем *Bash*); для некоторых примеров мы также используем Ruby. Если вы хотите следовать за нами, убедитесь, что у вас есть все, что нужно. При необходимости все это можно установить из репозитория вашего дистрибутива. Чтобы приступить к работе, читайте наш материал дальше. Для начала мы наскоро проверим, что все необходимое у вас есть.

Настройка вашего репозитория Git

```
$ git --version
git version 1.8.3
$ ruby --version
ruby 2.0.0p0 (2013-02-24 revision 39474)
```

В целях нашего урока, создадим директорию для использования в качестве экспериментальной среды, где мы можем проводить испытания:

```
$ mkdir ~/git-sandbox
$ cd ~/git-sandbox
$ git init
Initialized empty Git repository in /home/john/git-sandbox/.git
```

Последняя команда превращает нашу пустую директорию в репозиторий Git. Она создает директорию, которая обычно скрыта и именуется **.git**. Заглянув внутрь этой директории, вы увидите три файла и четыре директории:

```
branches/
config
description
HEAD
hooks/
info/
objects/
refs/
```

Если вы использовали репозиторий Git для какого-то своего проекта, вы, вероятно, уже знакомы с файлом **config**. Он содержит специальные опции настройки проекта. Вероятно, вы настраивали этот файл вручную или использовали инструмент настройки Git для обновления своих параметров. Файл **description** используется для описания репозитория, причем только опциональным web-интерфейсом, GitWeb; в остальном его можно игнорировать.

Еще один файл именуется **HEAD**, и он, вместе с директориями объектов и ссылок [**refs**], формирует важные ключевые части Git. В директории объектов Git хранит файлы, содержащиеся в репозитории. Refs и **HEAD** — указатели на специальные объекты. Понимание этих объектов и указателей станет ключом к пониманию работы Git.

Прочие директории — информационная [**info**], где содержится файл под названием **exclude**, который служит тем же целям, что и более известный файл **.gitignore**, и директория веток [**branches**]; современным Git они не используются, и их можно спокойно игнорировать.

Итак, что же такое объекты? Давайте создадим объект, чтобы это выяснить:

```
$ echo 'hello, world!' > hello.txt
$ git hash-object -w hello.txt
```

Скорая помощь



Системы контроля версий — Version Control Systems (VCS) иногда называют системами управления исходным кодом — Source Code Management (SCM).

работает

```
270c611ee72c567bc1b2abec4cbc345bab9f15ba
```

Hash-object — пример команды типа plumbing. Git не без специфического юмора называет свои низкоуровневые команды plumbing [канализация], а обычные команды уровня конечного пользователя — porcelain [сантехника]. В повседневном использовании конечный пользователь знает только команды porcelain, однако мы будем использовать некоторые команды plumbing, чтобы вывить реализацию Git.

Результат, который мы видим — это 40-символьный хэш SHA1 содержимого нашего файла. Объект файла называется blob; есть, как мы скоро увидим, также объекты tree, commit и tag. В этом нет ничего магического: вы можете с легкостью создать этот хэш сами, если пожелаете (здесь мы использовали Ruby):

```
$ irb
>> require 'digest/sha1'
=> true
>> file=File.open("hello.txt","r").read
=> "hello, world!\n"
>> Digest::SHA1.hexdigest "blob #{file.length}"/0#:file/"
=> "270c611ee72c567bc1b2abec4cbc345bab9f15ba"
```

Библиотека объектов Git работает как файловая система с адресацией по контенту, то есть хранящиеся объекты можно найти с помощью ключа, основанного на их контенте — это хэш SHA1. Заглянув внутрь `.git/objects`, вы увидите, что добавленный вами объект хранится в поддиректории с тем же названием, что и первые два символа его SHA1, и с именем файла, состоящим из остальных 38 символов.

```
.git/objects/27/0c611ee72c567bc1b2abec4cbc345bab9f15ba
```

С целью экономии места Git также сжимает свои объекты с помощью `Zlib`. Продолжая пример Ruby выше, вы также можете осуществить это сжатие самостоятельно:

```
>> require 'zlib'
=> true
>> Zlib::Deflate.deflate "blob #{file.length}"/0#:file/"
=> "\x9CK\xCA\xC9OR04a\xC8H\xCD\xC9\xC9xD7Q(\xCF\xCAIQ\xE4\x02\0P\xB5\x06\xD8"
```

Или можете вручную раздуть и извлечь данные вашего файла из сохраненного объекта файла:

```
>> object = File.open(".git/objects/27/0c611ee72c567bc1b2abec4cbc345bab9f15ba","r").read
```

Plumbing и porcelain

Повседневные команды Git высокого уровня porcelain реализуются низкоуровневым plumbing. Скажем,

```
git add hello.txt
будет выглядеть как
git update-index --add --cacheinfo 100644 \
```

Другие примеры таких альтернатив:

```
$ (git hash-object -w hello.txt) hello.txt
git commit -m "first commit"
становится
```

```
echo 'first commit' | git commit-tree \$(git
write-tree)
git checkout mybranch
становится
git update-ref HEAD mybranch
git reset
git commit
становятся
git write-tree
git update-ref
```

```
=> "\u0001K\xCA\xC9OR04a\xC8H\xCD\xC9\xC9xD7Q(\xCF\xCAIQ\xE4\u0002\0P\xB5\u0006\xD8"
>> inflated_object = Zlib::Inflate.inflate(object).match (/(//
CAPTION//
a-z]+) ([0-9+)(.*)/
=> #<MatchData "blob 14\0hello, world!" 1:"blob" 2:"14" 3:"hello,
world!">
>> inflated_object///CAPTION///3
=> "hello, world!"
```

Мы использовали регулярные выражения Ruby, чтобы извлечь из объекта его тип, размер и контент. Возвращается массив, где элемент 0 — целый объект, элемент 1 содержит его тип, элемент 2 содержит размер контента, а элемент 3 содержит актуальный контент.

Итак, мы создали объект из файла, но вам, возможно, интересно, что случилось с именем файла — наш файл именовался `hello.txt`, а в объекте это нигде не отражено. Имя хранится в индексе. Индекс — нечто вроде зала ожидания для файлов перед отправкой в репозиторий. Все, что будет отправлено, сначала должно быть добавлено в индекс, двоичную базу данных, хранящуюся внутри файла `.git/index` (который создается, когда в него впервые что-то добавляется). Мы добавили наш файл в индекс с помощью plumbing-команды `git update-index`:

```
$ git update-index --add --cacheinfo 100644 270c611ee72c567bc1b2abec4cbc345bab9f15ba hello.txt
```

»

SHA1: Магия за кулисами магии

Главное в магии Git — хэш SHA1. SHA — это акроним Secure Hash Algorithm, обозначающий семейство криптографических хэш-алгоритмов. SHA1 используется наиболее широко и производит 160-битное (20-байтное) значение хэша, которое выражено в виде 40-символьного шестнадцатеричного числа.

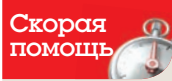
Git хранит объекты, используя их SHA1 в качестве ключа, и поскольку идентичные объекты будут иметь один

и тот же хэш, идентичный контент не дублируется, несмотря даже на то, что он может представлять собой разные вещи, например, два разных файла с одним и тем же контентом или два идентичных дерева директории в разных операциях подтверждения [commits].

Полный 40-символьный SHA1 не так просто запомнить, и большинство команд Git принимают сокращенный SHA1, коль скоро он остается уникальным. `git log`

`--abbrev-commit` отобразит сокращенные значения SHA1 для операций подтверждения, а `git --rev-parse <сокращенный-SHA1>` возвратит полный 40-символьный SHA1 сокращенного.

Теоретически возможно для разного контента иметь тот же самый хэш SHA1, что известно как конфликт. Но такое обычно считается настолько маловероятным, что этот потенциальный сценарий может игнорироваться.



Скорая помощь

`git cat-file -p <SHA1>` будет отображать контент любого объекта. **CODEINLINE-t** отображает его тип.

```
$ git ls-files --stage
100644 270c611ee72c567bc1b2abec4cbc345bab9f15ba 0 hello.txt
```

Обратите внимание, что в индекс мы добавили объект, а не файл. `update-index` добавляет объект с указанным SHA1 и данным именем и разрешениями (вам нужно здесь использовать полный 40-символьный SHA1). Для создания объекта из нашего файла мы использовали `plumbing`-команду `hash-object`, за которой следует `update-index`, и затем добавили его в индекс, однако обычный способ сделать это — использовать `porcelain`-команду `git add`, которая добавит файл за один шаг.

Перед этим команда `git status` отобразит наш файл как неотслеженный, но впоследствии он будет отображен как изменение, которое требует подтверждения. Итак, что такое операция подтверждения?

Операция подтверждения — это мгновенный снимок индекса плюс запись того, когда и кем был сделан этот мгновенный снимок. Индекс содержит все файлы, присутствующие в предыдущей операции подтверждения (при наличии таковой), плюс любые дополнения, ожидающие подтверждения, удаления или модификации. Мгновенный снимок содержится в объекте `tree`, и сделать его из текущего индекса очень просто:

```
$ git write-tree
24bbdca8b223aaa3384d78312f730c58492aa30a
```

Объекты Tree

Заглянув внутрь объекта `tree`, мы увидим, что он включает в себя `blob` SHA1, содержащий контент файла, а также его файловое имя, `hello.txt`. Также он перечисляет разрешения файла как 100644:

```
$ git ls-tree 24bb
100644 blob 270c611ee72c567bc1b2abec4cbc345bab9f15ba
hello.txt
```

Вот здесь вы и сталкиваетесь с ограничениями Git: он записывает разрешения файла как 100644 для обычного файла, 100755 для исполняемого или 120000 для символической ссылки. Если у вас файл с различными разрешениями — скажем, `rw-rw----` (660) — который вы проверяете в Git, его разрешения при проверке окажутся 644. Вы можете использовать что-то вроде `git-cache-meta`, чтобы справиться с этим. Загляните в <https://gist.github.com/andris9/1978266>.

Как и `blob`, объект `tree` может ссылаться на другие объекты `tree`. С такой рекурсией работает иерархия директории. `Plumbing`-команда `git write-tree` рекурсивно записывает объекты `tree` для всех файлов в индексе, и возвращает SHA1 самого верхнего объекта. Он затем может использоваться для доступа ко всем файлам в том виде, в котором они существовали на тот момент времени.

Объект `commit` располагается над объектом `tree` и служит для записи того, когда он делается, кем и зачем. Он содержит SHA1 для `tree` вместе с именем и адресом электронной почты того, кто делал подтверждение, и сообщением, содержащим примечания к нему. Создание объекта `commit` тоже дело несложное:

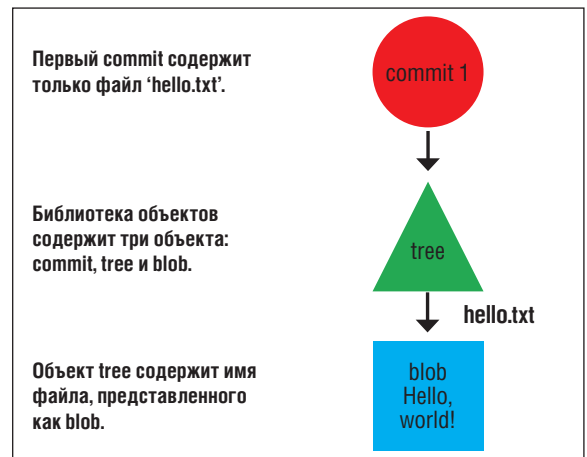
```
$ echo "my commit message" | git commit-tree 24bb
b4a8b6d97ec9e1cb6878b19b5eb6f9abe5aaeed8
```

Итак, у нас есть три объекта: `blob`, представляющий наш файл, `tree` и `commit` (см. рис. 1):

```
$ find .git/objects -type f
.git/objects/b4/a8b6d97ec9e1cb6878b19b5eb6f9abe5aaeed8
.git/objects/24/bbdca8b223aaa3384d78312f730c58492aa30a
.git/objects/27/0c611ee72c567bc1b2abec4cbc345bab9f15ba
```

Если вы следуете за нашим уроком, SHA1 вашего `commit`'а будет другим, потому что подтверждение содержит информацию о пользователе и времени, и в вашей системе она будет иной. Чтобы заглянуть внутрь объекта `commit`, используйте `git cat-file`:

```
$ git cat-file -p b4a8
tree 24bbdca8b223aaa3384d78312f730c58492aa30a
```



► Рис. 1. Библиотека объектов содержит три объекта после подтверждения одного файла.

```
author John <john@example.com> 1378821588 +0100
committer John <john@example.com> 1378821588 +0100my
commit message
```

Теперь наш файл подтвержден в Git, и к нему можно получить доступ, коль скоро известен хэш SHA1 этого подтверждения. Последняя часть процесса подтверждения устраняет необходимость запоминать SHA1 подтверждения посредством обновления текущей `head`, которая на него указывает. `Head` — это указатель на подтверждение, и это просто файл в `.git/refs/heads`, содержащий SHA1 этого подтверждения. Обновляемая `head` — это текущая `head`, как указано в файле `.git/HEAD`:

```
$ cat .git/HEAD
ref: refs/heads/master
```

Поэтому текущая `head` называется «мастер», и ее обновление до точки нашего подтверждения — это просто необходимость записать в нее SHA1 подтверждения:

```
$ echo "b4a8b6d97ec9e1cb6878b19b5eb6f9abe5aaeed8" > .git/refs/heads/master
```

Ветки

Вот так. Другое название для `head` — ветка. Системы SCM используют ветки для обеспечения параллельных путей разработки, позволяя кодовой базе время от времени расходиться и снова сходиться. Некоторые системы SCM реализуют ветвление посредством создания копий файлов проекта (что отнимает место и может быть долгим процессом), однако ветка в Git — не что иное, как указатель на подтверждение. Вот почему их часто именуют легковесными ветками. Термины `HEAD`, текущая `head` и текущая ветка синонимичны. `Porcelain`-команда `git branch` просто перечисляет файлы в `.git/refs/heads`, помещая звездочку позади того, к которому относится `.git/HEAD`. `Head` может также обновляться с помощью `plumbing`:

```
$ git update-ref refs/heads/master b4a8
или даже
$ git update-ref HEAD b4a8
```

Использование `update-ref` имеет преимущество возможности принять урезанный SHA1 и произвести проверку, чтобы убедиться, что объект существует и является действительным `commit`. Если вы измените текущую `head` таким образом, вам также понадобится перестроить индекс, чтобы отразить эту ветку — чего можно добиться с помощью `git reset`. Обычно вместо этого применяют `porcelain`-команду `git checkout`, и она позаботится об индексе.

Еще один момент, на который ссылаются — это `tag`. Он похож на ветку, с той разницей, что `tag` всегда указывает на тот же самый `commit`, тогда как ветвь обновляется для каждого `commit`.

Есть два вида tag'ов: легковесный и аннотированный. Легковесный tag — это просто файл в `.git/refs/tags`, содержащий SHA1 объекта, на который он указывает:

```
$ git update-ref refs/tags/mytag b4a8
```

Аннотированный tag пишет объект tag и указывает ссылку на него вместо commit. Они создаются так:

```
$ git tag -a mytag b4a8 -m "tag message"
```

Обычно tag указывает на commit, но может указывать на любой вид объекта. Например, вы можете создать tag на специальный blob, если вам это нужно.

Третий тип ссылок — удаленная ссылка на remote. Remote — это еще один репозиторий, который вы отслеживаете, а внешние ссылки — это ссылки на ветки и tag'и в этом репозитории. Они обновляются, когда вы обращаетесь к remote (с помощью `pull`, `fetch` или `push`). Внешние ветки нельзя проверить, но вы можете создать собственную ветку:

```
$ git remote add bigproject http://example.com/bigproject.git
```

```
$ git fetch bigproject
```

```
remote: Counting objects: 214, done.
```

```
remote: Compressing objects: 100% (144/144), done.
```

```
remote: Total 214 (delta 58), reused 194 (delta 38)
```

```
Receiving objects: 100% (214/214), 108.18 KiB | 0 bytes/s, done.
```

```
Resolving deltas: 100% (58/58), done.
```

```
$ git checkout -m working bigproject/master
```

```
Branch working set up to track remote branch master from bigproject. Switched to a new branch 'working'
```

По вызову remote его объекты копируются в вашу локальную библиотеку объектов в `.git/objects`. Это не влияет на ваши собственные объекты, потому что хэши SHA1 уникальны. Ссылки на любые внешние ветви записаны в `.git/refs/remotes`, а tag записаны в `.git/refs/tags`. Если у вас уже есть tag с тем же именем, что и тот, который используется в remote, ваш tag не переписывается, но вы не получаете локальную копию tag в remote.

Hello — так всем hello

Чтобы вполне оценить структуру объектов, которые создает Git, мы должны все немного усложнить, поэтому мы расширим наш пример *Hello, world!*, чтобы продемонстрировать, как это приветствие может выражаться в разных частях мира. Наш пример показывает его на испанском для Испании и на английском — для США и Великобритании. Расщедрившись, мы сохраним наши приветствия в директориях для разных частей света.

```
$ mkdir europe north-america
```

```
$ mv hello.txt europe/uk-hello.txt
```

```
$ cp europe/uk-hello.txt north-america/us-hello.txt
```

```
$ echo 'hola, mundo!' > europe/es-hello.txt
```

```
$ git add --all
```

```
$ git commit -m "second commit message"
```

Как видно на диаграмме (рис. 2), теперь мы получили структуру директории с тремя файлами. После подтверждения наших изменений мы добавили только один blob для библиотеки объектов для перевода на испанский — английское приветствие не изменилось с нашего первого подтверждения, и сейчас на этот исходный blob ссылаются в трех случаях: для исходного файла `hello.txt` в первом подтверждении, и `uk-hello.txt` и `us-hello.txt` во втором подтверждении.

Часто и бурно обсуждается вопрос о том, хранит ли Git разность полученных результатов (имеется в виду только разница между самым последним и предыдущим подтверждениями). Те, кто использовал другие системы SCM, часто удивляются тому, что Git хранит новую копию всего репозитория для каждого подтверждения. Однако все не так плохо, как кажется, потому что файл (или даже дерево [tree]), который остается неизменным между двумя подтверждениями, будет иметь один и тот же SHA1, и, следовательно, тот же объект, устраняя возможное дублирование.

Отдельный заголовок

Вы можете проверить любое подтверждение, даже достаточно старое, даже если в нем нет заголовка-head, непосредственно указывающего на него. Это даст вам то, что называется «отдельный заголовок», и это просто означает, что вы проверили заголовок, на который нет ссылки в `.git/refs/heads`. Если у вас есть отдельный заголовок, то `.git/HEAD` будет содержать SHA1 вместо ссылки. Вы по-прежнему можете вносить изме-

нения и подтверждать их. Но вы рискуете потерять все изменения в следующий раз, когда запустится сборщик мусора Git, потому что подтверждения на отдельный заголовках не имеют ссылок. Чтобы этого не случилось, после проверки настройте новую ветку: `git checkout -n new_branch_name`.

Если вы подтвердили отдельный заголовок и проверили что-то еще, то `git reflog` может помочь вам снова его найти.

Однако остается тот факт, что две версии файла с разницей в один символ будут храниться совершенно неэффективно — как два отдельных файла (поскольку их хэши SHA1 будут отличаться). Git удаляет дубликаты, уплотняя объекты; это процесс, который сжимает объекты и использует дельта-компрессию. Объекты в `.git/objects` именованы неприкрепленными [loose] объектами, и их уплотнение удаляет дубликаты, созданные из-за небольших изменений. Уплотнение производится периодически мусорщиком Git [garbage collector], который вы также можете вызвать по требованию с помощью `git gc`.

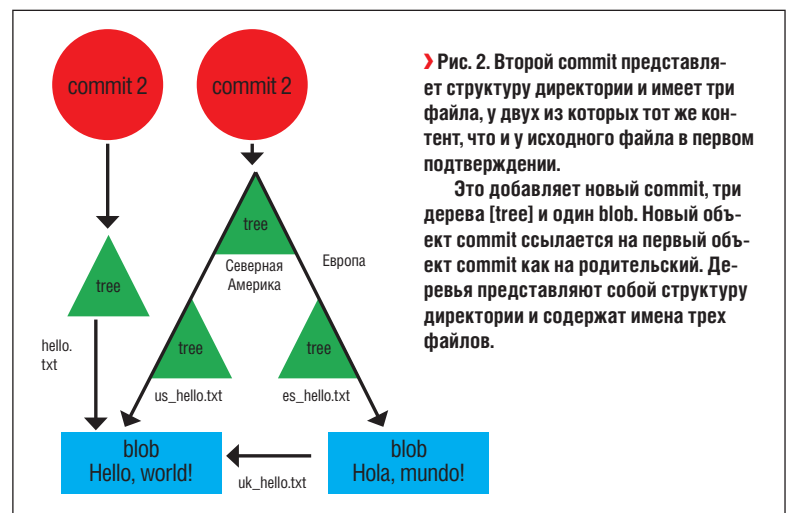
Уплотните

Сборщик мусора периодически проходит по дереву объектов в индексе и по каждой ссылке (веткам и tag'am), и отмечает каждый достижимый объект. Он удаляет все недостижимые объекты и сжимает все оставшиеся подтвержденные объекты в пакеты. Неподтвержденные объекты (в индексе) остаются неприкрепленными. После уплотнения вы найдете несколько новых файлов в `.git/objects`, а ваши исходные файлы объектов будут удалены. Новые файлы — это `packfile` и `index`. Содержимое всех удаленных объектов теперь находится в `packfile`, а `index` содержит смещения [offsets] объектов в `packfile`.

Декодирование пакетов находится за пределами тематики данной статьи, но вы можете получить доступ к любому объекту в пакете через индекс. Мы использовали `git verify-pack` или `git show-index` для отображения содержимого пакетов. Впоследствии к любому уплотненному объекту можно получить доступ через его SHA1 и `git cat-file`.

Более подробную информацию вы найдете в книге *Pro Git*. Ее можно читать бесплатно онлайн на <http://git-scm.com/book>. **LXF**

► В нашем втором commit, три файла представлены одним blob'ом!



► Рис. 2. Второй commit представляет структуру директории и имеет три файла, у двух из которых тот же контент, что и у исходного файла в первом подтверждении.

Это добавляет новый commit, три дерева [tree] и один blob. Новый объект commit ссылается на первый объект commit как на родительский. Деревья представляют собой структуру директории и содержат имена трех файлов.



ЧАСТЬ 4

Blender: ВнедриМ

Андрей Прахов совершенствуется в обманах зрения, эксплуатируя заветные секреты дизайнеров. Маска, я тебя знаю!



Наш эксперт

Андрей Прахов
Участник нескольких игровых проектов, представитель СМИ, автор первой русской книги по Blender: «Blender. 3D-моделирование и анимация. Руководство для начинающих».

Есть нечто завораживающее в том, как герой какого-нибудь популярного фильма, с большущей пушкой в руках, крошит налево и направо монстров. Даже зная о технологиях использования Green Screen и Motion Tracking, невольно задаешься вопросом: а как? Теперь, конечно, прочитав предыдущие статьи на эту тему, вы уже в курсе за кадром стороны некоторых киношных эффектов. Но есть еще несколько моментов, заслуживающих рассмотрения...

Трекинг объекта

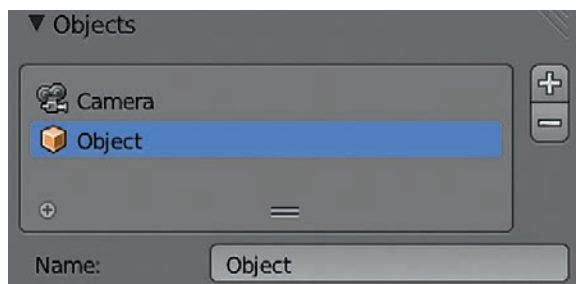
Система Motion Tracking, предлагаемая Blender, уже не является чем-то недостижимым для понимания. Даже в домашних условиях, имея на руках бытовую камеру, можно добиться потрясающих успехов. YouTube буквально кишит такими самодельными клипами. Думаю, вы некоторые из них видели. Особенно популярно «прицепить» на голову страшную маску или размахивать виртуальным мечом. И зрелищно, и просто в исполнении.

Как вы помните, основой трекинга камеры является использование специальных меток. Чем больше таких меток, тем лучше, но, главное, их должно быть не менее 8 в каждом кадре. Такое же правило действительно и для трекинга конкретного объекта.

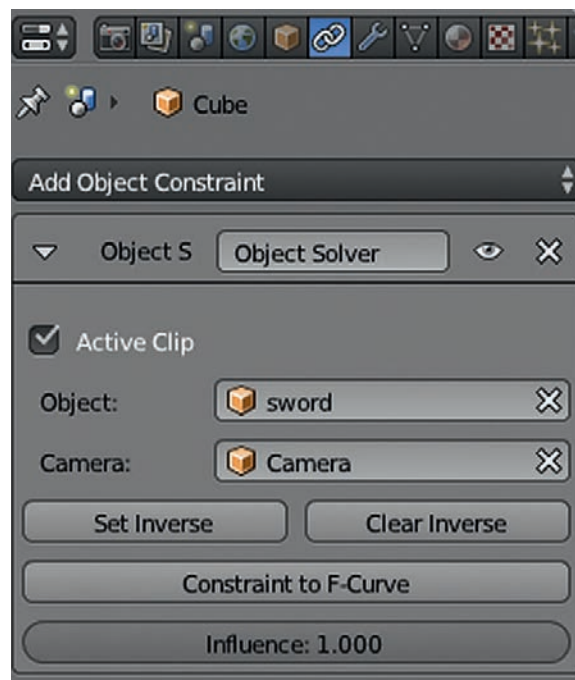
Возьмем, к примеру, съемочный эпизод типичной городской улицы. Движение самой камеры отслеживается благодаря неподвижным маркерам в сцене (камни на дороге, рисунки на стенах домов, светофоры и т.д.). Но есть и внутренние перемещения — это спешащие по своим делам пешеходы, машины, животные. Такое движение носит название объектного и может быть отслежено дополнительно к движению камеры.

Если создать в Blender новый проект Motion Tracking, то программа вначале предложит отследить движение камеры. Даже если камера в съемках абсолютно неподвижна, стоит выполнить фиктивный трекинг. Как это делается, уже обсуждалось в предыдущей статье. А вот для добавления других объектов трекинга имеется специальное окошко в правой части Movie Clip Editor (рис. 1).

По умолчанию здесь присутствует лишь один пункт с характерным названием Camera. Кнопкой «плюс» можно добавить новый объект. Заметьте, что главное окно редактора при этом очистится от всех ранее установленных маркеров, да и левая колонка опций несколько изменится. Blender требует для каждого нового объекта свой набор меток.



► Рис. 1. Здесь добавляется нужное количество объектов для трекинга.



► Рис. 2. Привязка системы трекинга к объекту.

В принципе, создание маркеров и выполнение собственно трекинга ничем не отличается от камерного, за одним исключением: Blender не умеет автоматически привязывать созданный трекинг к трехмерному объекту в сцене. Да и неудивительно — откуда программа может это знать?

После выполнения просчета кнопкой Object Motion (аналог ранее рассмотренной функции Camera Motion) в окне 3D View должны появиться Dumpy-объекты. Именно они автоматически привязываются к выполненному трекингу. По сути, эти объекты служат только для контроля качества трекинга и не выводятся при рендере сцены, а вот привязка самого трехмерного объекта к результатам трекинга осуществляется с помощью специального Ограничителя [Constraint] — Object Solver (рис. 2).

Использовать этот ограничитель очень просто. Сначала выделяется в сцене нужный трехмерный объект. Затем к нему добавляется ограничитель Object Solver (см. рис. 2). А уже в его настройках выбирается выполненный ранее трекинг для самого объекта и камеры.

После этих нехитрых манипуляций ваша модель будет привязана к движению в кадре. Только не забудьте совместить координаты и вращение трехмерного объекта и dumpy, что появились после просчета сцены (рис. 3).

Совершенствуемся в обмане

Посмотрите на рис. 3. Как вы думаете, можно ли разместить трехмерный объект между человеком и деревьями? Только не предлагайте решение с хромакей, ведь девушка и лесной фон уже запечатлены видеокамерой на одной ленте!

В ТЕЛЕВИДЕНИЕ



Есть среди секретов дизайнеров такой прием, как «маскирование». Это когда часть изображения отделяется от основного фона и является уникальной единицей. Нечто подобное делается при хромакей. Только в данном случае указывается произвольный контур (маска) вокруг требуемого изображения, а все остальное устанавливается в прозрачность. Можно сделать и наоборот, когда область маски является прозрачной, а фон остается. Великолепный прием, знакомый всем дизайнерам; но при работе с видео появляются определенные сложности с движением в кадре. Ведь контур маски должен гибко реагировать на все изменения картинки. Вот тут-то опять всплывает на поверхность необходимость трекинга изображения.

В случае с *Blender*, работа с масками встроена в систему Motion Tracking. И работает этот механизм на удивление гладко.

Допустим, необходимо, чтобы некоторый объект пролетел между головой девушки и лесным фоном. Соответственно, нужно создать маску вокруг головы. Делается это очень просто.

В окне Movie Clip Editor имеется меню *Mode*, где можно выбрать режим работы системы трекинга. По умолчанию там значится пункт *Tracking*. Итак, первым делом нужно выбрать в этом меню режим *Mask*.

Маска — это по сути обычная кривая (сплайн), где ее формой управляют с помощью контрольных точек. Они, в свою очередь, добавляются щелчком левой кнопки мыши при нажатой клавише *Ctrl*. Причем по мере установки новых точек *Blender* постарается скорректировать кривую так, чтобы она оптимально облегла требуемый рисунок. Разумеется, такое не всегда возможно. Поэтому придется вручную корректировать сплайн. Принцип работы с ним тот же, что и с обычными трехмерными объектами. Вы можете перемещать, вращать отдельные вершины или целые группы, удалять ненужные звенья. Главное — следить за тем, чтобы кривая не перекручивалась и не создавала замкнутые петли (рис. 4).

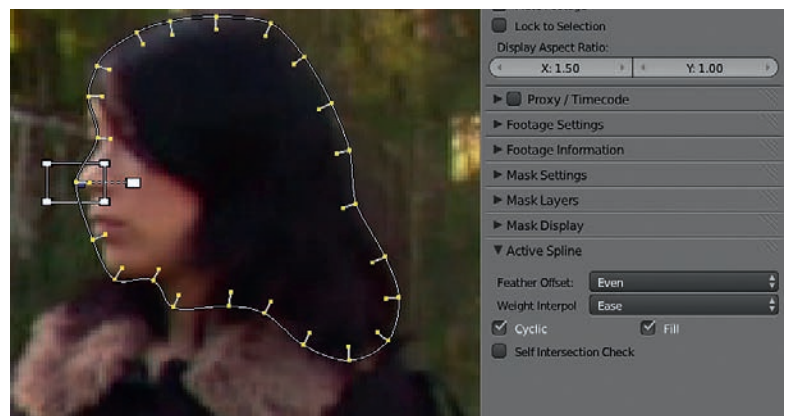
Крайние точки кривой необходимо разместить так, чтобы программа могла замкнуть сплайн. Для этого включите опцию *Cyclic* в настройках *Active Spline* (см. рис. 4).

По окончании работы с контуром нужно привязать созданную кривую к системе трекинга:

- 1 Перейти в режим *Tracking* (меню *Mode*).
- 2 Создать новую группу маркеров в списке объектов.
- 3 Добавить метку в нужном месте (в данном примере маркер отслеживает перемещения носа девушки). Здесь нет ограничений на количество меток: хоть одну, хоть десять.
- 4 Выполнить просчет трассы.
- 5 Перейти в режим *Mask*. Удерживая клавишу *Shift*, щелкнуть правой кнопкой мыши по контуру и нажать *Ctrl+P*. То есть, таким образом объект маски привязывается к маркеру трекинга.

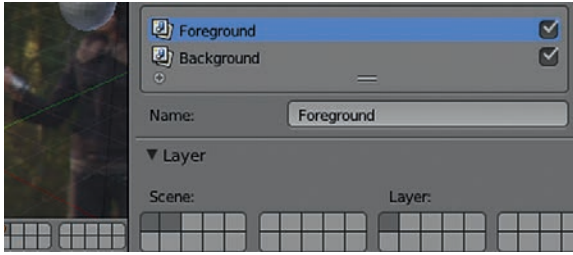
После указанных манипуляций контур маски будет двигаться вслед за маркером. Причем *Blender* опять-таки постарается

» Рис. 3. Отчетливо видно, что девушка держит в руках обычную палку с разноцветными метками, а вот уже справа почти что меч.



» Рис. 4. Маска и маркер трекинга вокруг головы.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



» Рис. 5. Параметры рендера.

оптимизировать форму сплайна так, чтобы он облегал нужный рисунок. Но ничто не мешает вам в требуемом кадре вручную скорректировать маску. *Blender* все запомнит.

Однако это только часть работы. Ведь еще нужно расположить все по слоям, настроить ноды композиции — в общем, создать целый конвейер!

Для рассматриваемого примера оптимально будет распределить объекты по двум слоям. В первом пусть находятся камера и вспомогательные объекты трекинга, а во втором — нужная модель (рис. 5).

На рис. 5 показана часть настроек рендера. Обратите внимание, что под именем *Foreground* скрывается первый слой сцены, а *Background* содержит второй слой (на рисунке этого не видно).

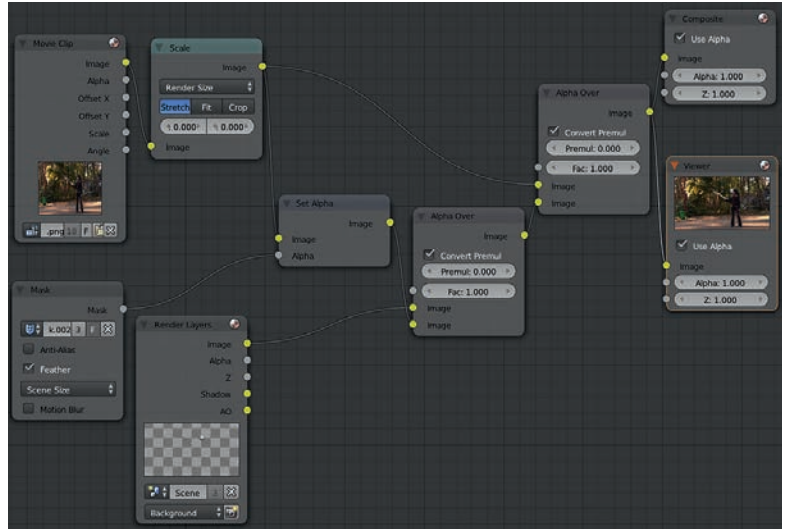
Теперь перейдем к созданию конвейера нод. В действительности, ничего сложного в этом нет. Давайте пройдемся по нодам рис. 6 и разберемся, что же тут происходит.

В качестве источников в этом конвейере служат ноды:

- » **Movie Clip** Вывод видеозображения
- » **Mask** Вывод созданной маски.
- » **Render Layers** Вывод конкретного слоя *Background* с моделями.

Представим сцену в виде слоеного пирога, где самый нижний слой будет занимать фоновое видео. Промежуточным слоем идет трехмерная модель, а самый верхний занимает маска головы.

Посмотрите внимательно на рис. 6. Видеоизображение с ноды *Movie Clip* подается на ноду *Scale*. Таким образом можно



» Рис. 6. Конвейер нод.

проконтролировать правильность пропорций показа видео в соответствие с его характеристиками. Затем с помощью узла *Set Alpha* устанавливается альфа-канал, где прозрачностью является сама маска, а уже смикшированный сигнал объединяется со слоем *Background* с помощью ноды *Alpha Over*. Получается, что в данный момент маска будет располагаться поверх трехмерного объекта. Осталось только наложить этот сигнал на фоновое видео опять-таки с помощью очередного *Alpha Over* и подать полностью готовый сигнал на вывод рендера (рис. 7).

Указанная схема конвейера вполне работоспособна при заявленных стартовых условиях. Но ее можно подкорректировать, для получения более качественного изображения. Так, маска выполняет весьма грубую обрезку лица. Сгладить, а точнее размыть границы можно с помощью специальных фильтров.

Все зависит от поставленной задачи, но можно с полной уверенностью заявить: инструменты, предоставляемые *Blender*, позволяют с высочайшим качеством производить сложный композитинг сцены! **LXF**



» Рис. 7. А это уже результат наших экспериментов. Слева — готовый рендер кадра, где видно, что сфера проходит позади головы девушки, а вот справа — реальное положение вещей в окне *3D View*.

» **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ВИРТУАЛИЗАЦИЯ: ЭКОНОМИЯ НА СТОИМОСТИ IT-ИНФРАСТРУКТУРЫ ДО 90%



Единая инфраструктура
на базе свободного
программного обеспечения

Нет лицензионных платежей —
расходы только на внедрение
и техническую поддержку

Минимальные затраты
на оборудование
за счет виртуализации



[www.linuxcenter.ru/shop/
linux-software/office/kitezh](http://www.linuxcenter.ru/shop/linux-software/office/kitezh)

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

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 Проверку параметров SMART жесткого диска
- 2 Хитроумный код
- 3 Перенаправление вывода команды
- 4 Фотографирование на Raspberry Pi
- 5 Решение проблем с драйвером
- 6 Проблемы с разбиением диска
- 7 Wi-Fi

1 Поумнеем со SMART

В Пожалуйста, подскажите, как проверить, в порядке ли жесткий диск или он выходит из строя? У меня Ubuntu 13.04.

Narendra, с форумов

Если у Вас есть мельчайшее подозрение, что жесткий диск может выйти из строя, сразу сделайте резервную копию. Когда данные будут в безопасном месте, можно проверить диск, обычно с помощью S.M.A.R.T (далее для удобства будем называть это SMART). Хотя это сокращение напоминает название криминальной организации из пародии на шпионские фильмы шестидесятых, на самом деле оно означает "Self-Monitoring, Analysis and Reporting Technology [Технология самоконтроля, анализа и отчетности]".

С помощью этой технологии жесткие диски могут отслеживать свое состояние и сообщать о нем операционной системе до возникновения любых проблем. Существует пара условий использования SMART. Во-первых, Ваш диск должен ее поддерживать — все современные диски делают это. Во-вторых, она должна быть включена в BIOS; так что перезагрузитесь в BIOS и проверьте.

Также Вам понадобится установить у себя пакет *smartmontools*. В нем есть две программы: *smartd* — демон, который работает в фоне, отслеживая состояние дисков и сообщая Вам обо всех проблемах, и *smartctl* — пользовательская утилита проверки дисков. Во-первых, убедитесь, что SMART поддерживается Вашим диском, выполнив команду

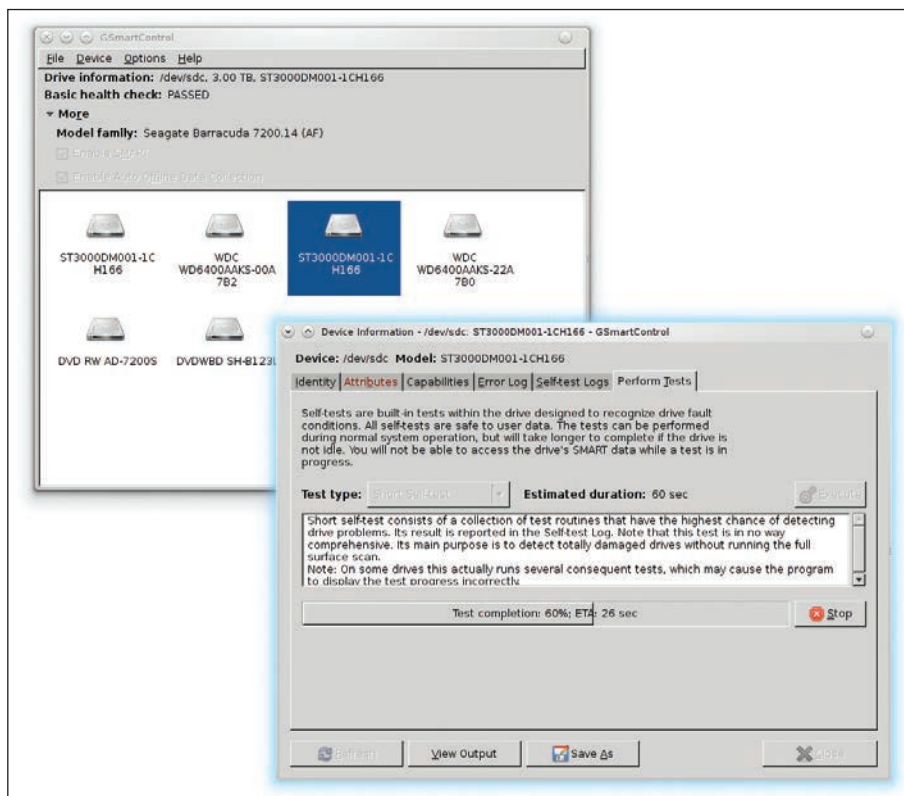
```
smartctl -i /dev/sda
```

Она выведет разную информацию о диске — в конце должно быть следующее:

```
SMART support is: Available - device has SMART capability.
```

```
SMART support is: Enabled
```

Если утилита сообщает, что данная возможность недоступна [unavailable], Вы не сможете проверить диск с помощью SMART; если отключена



➤ GSmartControl предоставляет удобный интерфейс для утилит самопроверки диска SMART.

[disabled], Вы не включили ее в BIOS. Для получения краткой информации о состоянии диска скомандуйте

```
sudo smartctl --health /dev/sda
```

Существует несколько тестов, которые можно выполнить с диском. Короткий тест можно запустить прямо в работающей системе, и обычно на его выполнение уходит несколько минут:

```
sudo smartctl --test=short /dev/sda
```

Утилита сразу вернет Вас в командную строку, так как тест выполняется в фоне. Текущее состояние выполняющихся тестов и результаты завершенных можно просмотреть командой

```
sudo smartctl --all /dev/sda
```

Если Вы хотите полностью удостовериться, запустите длинный, более тщательный тест. Также есть параметр `--captive`, с его помощью тест можно заставить выполняться не в фоновом режиме; но этим параметром нельзя пользоваться, если смонтирован хотя бы один раздел — он подходит только для Live CD.

Программа *smartd* выполняется в фоновом режиме, регулярно проверяя диски и сообщая об обнаруженных ошибках по электронной почте. Выполняемые тесты и адрес электронной

почты нужно задать в `/etc/smartd.conf`. Файл содержит исчерпывающие комментарии и примеры. Существует и графическая альтернатива *smartctl*, под названием *GSmartControl*, в которой общие функции *smartctl* реализованы в графическом интерфейсе.

2 Как с долларами?

В Я искал в Интернете программу для генерации большого двоичного файла для программы, которую я тестирую, и наткнулся на следующий код:

```
dd if=/dev/zero of=10g.img bs=1000 count=0 seek=${1000*1000*10}
```

Я знаю, что делает команда `dd`, но что означает \$ в параметре `seek`? Запустил скрипт, но не получил ничего. Как может счетчик (`count`) равняться 0?

PK Fox

Знак доллара с квадратными скобками заменяется результатом вычислений, содержащихся в скобках. Попробуйте выполнить команду:

```
echo ${1000*1000*10}
```

хотя чаще для этого используются двойные круглые скобки:

```
echo $(( 1000*1000*10 ))
```

поэтому Вы не нашли бы их на map-странице *Bash*. В скобках можно пользоваться переменными:

```
X=10
echo $(( X * 1024 * 1024 ))
```

Эта конструкция всего лишь представляет 10 Гб в более читаемом виде. При таком запуске команды **dd** создается разреженный [sparse] файл [файл, в котором последовательности нулевых байтов заменены на информацию об этих последовательностях, — прим. пер.]; чаще используется **count=1**, но **0** тоже работает. Параметр **seek** указывает для **dd** стартовую позицию, поэтому все предыдущее пространство должно быть создано. Разреженный файл использует только то место, которое используется находящимися в нем данными, поэтому хотя команда

```
ls -l 10g.img
```

покажет файл большого размера, команда

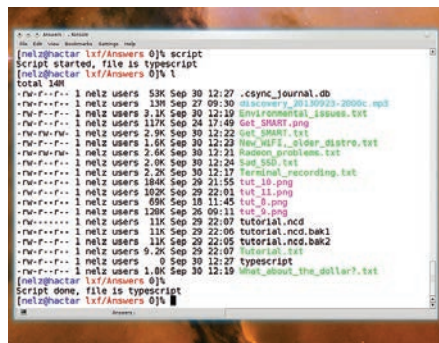
```
du -h 10g.img
```

покажет место на диске, которое файл занимает на самом деле. Этот прием часто используется при создании диска виртуальной машины. Размер, отображаемый **ls**, останется постоянным, а занимаемое пространство будет расти по мере использования файла. Такой виртуальный диск будет работать быстрее по сравнению с обычным файлом, который растет по мере использования, что часто приводит к большой фрагментации.

Не уверен, насколько это Вам подходит, так как в файле не будет данных, пока Вы их не добавите, но если Вам нужен файл-контейнер, который заполнится позже, как и в случае с образами дисков виртуальных машин — это хороший подход.

3 Запись терминала

В Я порезвился с птар, и это заняло какое-то время. Если я запускаю команду в терминале и получаю какой-то вывод, можно ли



➤ **Запустите script, и ввод и вывод терминала будут записываться, пока вы это не остановите.**

перенаправить этот вывод куда-то после того, как команда была выполнена, или придется еще раз запустить команду с каналом?

Fat_Tuesday, с форумов

Вы спрашиваете, как закрыть дверь конюшни, когда лошадь удрала, но на самом деле несколько вариантов есть, в зависимости от того, каким терминалом Вы пользуетесь. В *Konsole* из KDE (на него стоит обратить внимание, даже если Вы не пользуетесь KDE) есть пункт **Save Output As** [Сохранить вывод как...] в меню **File** [Файл]. Он сохраняет весь буфер прокрутки в файл. В *Gnome-terminal* на данный момент такой возможности нет. Если Вы не в X и пользуетесь виртуальной консолью, может помочь команда **setterm**:

```
setterm -dump -file output.txt
```

Она, или соответствующая ей команда **append**, записывает вывод консоли в файл, но записывается только видимая часть буфера, поэтому команда не слишком полезна.

Лучшим вариантом для Вас будет приучиться при необходимости перенаправлять вывод команды прямо в файл. Если одновременно нужно его видеть, можно воспользоваться **tee**,

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**.

которая перенаправляет свой ввод и в стандартный вывод, и в файл:

```
somescmd | tee somefile.log
```

Есть и еще один вариант (как всегда в Linux), и он называется *script*. Вы запускаете эту команду один раз в терминале, и весь вывод с этого момента сохраняется в файле. Эта программа запускает еще одну оболочку в той, в которой Вы работаете, ➤



Коротко про...

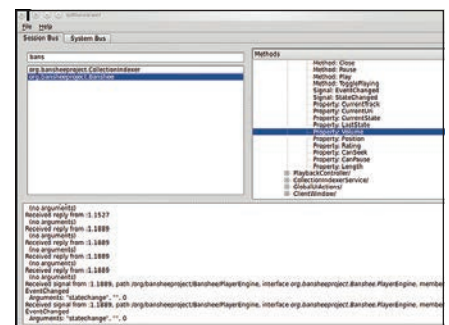
D-bus

D-Bus, или Desktop Bus (настольная шина) — низкоуровневая система межпроцессного взаимодействия, используемая в Linux и других операционных системах POSIX. С ее помощью программы могут отправлять сообщения друг другу. До нее использовались по меньшей мере две несовместимые системы: DCOP в KDE и Волобо в Gnome. Архитектура DCOP в значительной степени повлияла на D-Bus, но D-Bus не зависит от рабочего стола, несмотря на Desktop в названии. В настольных компьютерах в общем используются две шины: системная шина, используемая программами **root**, такими как демоны, ПО определения

устройств и подобные им, и сессионная шина, используемая рабочим столом.

Пример использования D-Bus можно увидеть в вопросе «Banshee молчит» в **LXF156**, но ее возможности гораздо шире, и многие программы, от программ определения устройств до системных оповещений, незаметно от пользователя применяют ее. Команды для управления шиной из оболочки с длинными именами сервисов и путями до объектов могут показаться громоздкими, но это необходимо для того, чтобы все программы могли получить доступ к D-Bus без возникновения конфликтов. Самый простой способ познакомиться с шиной — просматривая дерево шины командой **qdbusviewer**, выполняя различные команды и наблюдать за результатом. Если вы хотите получить доступ к интерфейсу программы на D-Bus с помощью

скрипта, самый простой вариант — **qdbus**. Также есть **dbus-send**, но он работает на более низком уровне, и чтобы что-то в нем сделать, придется сначала почитать справку в **man**.



➤ **Секреты D-Bus раскрыты!**

получает контроль над входным и выходным потоками и сохраняет все в файл до тех пор, пока Вы не выйдете из этой оболочки командой Ctrl+D, после чего Вы вернетесь в обычную оболочку, и визуальной разницы не будет, если не считать файла `typescript`, содержащего весь вывод прошедшего сеанса. Вы бы и не заметили, что были в другой оболочке — так как она наследует Ваше окружение — если бы не одна деталь. Так как Вы находитесь в «подоболочке», у Вас другая история команд, поэтому после выхода из `script` при нажатии стрелки вверх появятся команды, которые запускались до `script`, а не во время ее работы. Если Вам нужны эти команды, они сохранились в файле `typescript` вместе с их выводом.

4 Проблемы со средой запуска

В У меня Raspberry Pi с очень удобным дисплеем Adafruit, и на нем установлен Arch. Я написал скрипт на Python, который запускается во время загрузки и с помощью скрипта `gphoto2` делает последовательность фотографий на подключенной цифровой камере Nikon.

Скрипт позволяет указать на дисплее количество фотографий и время между снимками.

Для сообщения камере, что нужно сохранять изображения на SD-карте камеры, используется команда `gphoto2 -set-config-value`. Если я запускаю скрипт в Python (это нужно делать с помощью `sudo`, чтобы запустить модули Adafruit), он выполняется корректно и сохраняет изображения в камеру. Если запускать его через `systemd` с сервисным скриптом, ссылающимся на тот же самый файл, при загрузке или командой `sudo systemctl start gphoto.service`, он выполняет команду для задания конфигурации, но потом, похоже, игнорирует ее.

Скрипт выполняется без ошибок, но изображения не сохраняются. Я добавил в скрипт журналирование, которое показывает, что команды выполняются.

Пытался запускать его на ноутбуке, но происходит то же самое, и я думаю, что проблема в `systemd` (или в том, как я им пользуюсь), а не в Pi. **Фил Майотт (Phil Myott)**

О Проблема связана не с `systemd`, а со средой, в которой выполняется скрипт. Когда я попытался запустить скрипт с этими командами из `cron`, то получил тот же самый результат. Между запуском скрипта или программы из демона, такого как `systemd`, и из оболочки есть разница. Когда Вы открываете сеанс оболочки, создается специальная среда — среда, которой нет при запуске из демона, даже под тем же самым пользователем. Проблема здесь не в том, что Вы запускаете скрипт из `systemd`, а в том, что Вы запускаете его не из сеанса оболочки.

Решение простое, хотя может показаться малость закрученным. Если Ваш скрипт изначально запускался в оболочке, можно загрузить профиль в первых строках скрипта одной из команд:

```
source /root/.bash_profile
source /root/.profile
```

в зависимости от того, в какой оболочке Вы запускаете скрипт. Команда `source` выполняет содержимое файла профиля в текущем сеансе оболочки. Обычно при выполнении скрипта в оболочке он выполняется в подоболочке, и текущая оболочка остается нетронутой — это можно проверить, запустив скрипт с единственной командой `cd /some/path` и посмотрев на текущий каталог в оболочке после его запуска. Так как у Вас скрипт на Python, нужно запускать его в сеансе оболочки, содержащем правильное окружение, что можно сделать с помощью скрипта-обертки:

```
#!/bin/bash
source /root/.bash_profile
/home/pi/timelapse.py
```

Скрипт устанавливает правильное окружение и затем запускает скрипт Python, который наследует это окружение. Это можно сделать в служебном файле, не помещая команды в отдельный скрипт:

```
ExecStart=bash --login -c "python /home/pi/timelapse.py"
```

Правда, я не проверял этот вариант, и мне ближе вариант со скриптом-оберткой, так как с ним все проще наблюдать и отлаживать за счет лишь одного дополнительного файла.

5 Пересадка SSD

В У меня Asus 701 Eee PC с жестким диском в 4 Гб, которым я пользовался во время путешествий, но с ним настали проблемы.

Я пытался установить другую ОС (что я делаю время от времени), но все пошло не так. Файлы я смог восстановить, но теперь на нетбуке вообще нет ОС. Я сделал USB-брелок с `Gparted` и выполнил команду `fdisk -l` для проверки диска — с виду все нормально. Потом я запустил `Gparted` и застрял: не смог понять, как настроить разделы. Наконец, я попробовал установить другую ОС, но получил сообщение “disk error, press any key to restart [ошибка диска, нажмите любую клавишу для перезагрузки]” и ничего не произошло.

Кен Мерфи (Ken Murphy)

О `Fdisk` на самом деле не проверяет диск — она считывает только таблицу разделов в первых 512 байтах устройства. При установке дистрибутива Вам не нужно беспокоиться о разбиении диска на разделы — установщики сделают это за Вас. В них обычно есть вариант «Использовать весь диск» или аналогичный. Выберете этот вариант, и установщик удалит все существующие разделы и создаст нужные ему. Если хотите, можете удалить существующие разделы с `Gparted`, но это не обязательно. С диском в 4 Гб особых вариантов разбиения не придумаешь — создайте раздел подкачки, а остальное оставьте корневой файловой системе. Опять же, любой установщик дистрибутива сделает это за Вас.

Ее PC был дешевым и приятным устройством, но Вашему 701, надо полагать, уже немало лет. У меня была более поздняя модель 901, и ее SSD-диск отказал несколько лет назад. В Ее PC не используются ныне доступные улучшенные SSD-диски, с рассредоточением износа,

увеличивающим срок их службы, поэтому я бы не удивился, если Ваш диск просто-напросто вышел из строя. Диск устанавливается внутри ноутбука, и его замена экономически неоправдана, но сегодня можно купить очень дешевые карты памяти, которые подключаются по USB. У меня диск в 16 Гб, который выступает всего на 7 мм и стоит менее 10 фунтов. Можно установить дистрибутив на такой диск, продлив срок жизни нетбука и увеличив место на диске.

6 Драйвер для Radeon

В Я обновил систему по предложению Ubuntu 13.04. После перезагрузки компьютером стало невозможно пользоваться: фон заполнил весь экран, и ни верхнего меню, ни иконок не стало видно. Так как на этом компьютере у меня были проблемы с монитором во всех ОС Linux, я попробовал перезагрузить систему, заменив “`nomodeset`” на “`quiet splash`”, как прежде. После этого я загружал коммерческие драйверы для своей видеокарты. На сей раз трюк не сработал. Мне пришлось очистить весь жесткий диск и восстановить его содержимое из резервной копии с помощью Clonezilla. У меня есть более старый компьютер, и 13.04 на нем прекрасно работает!

У проблемного компьютера процессор i3-3220, 4 Гб RAM и видеокарта Radeon HD6450. Можете ли вы сказать, аппаратная ли это проблема?

Брайан Митчелл (Bryan Mitchell)

О Это больше похоже на проблему с драйвером. Вы сказали, что загружали коммерческий драйвер, но в Ubuntu 13.04 есть открытый драйвер Radeon, поддерживающий Вашу видеокарту. Если у Вас установлены оба драйвера, то это создает немало шансов на неприятности. Лучше всего пользоваться открытым драйвером — он поддерживает трехмерную графику для Вашей карты и лучше поддерживается сообществом Ubuntu. Для этого удалите коммерческий драйвер `fglrx` из своей системы и переустановите еще несколько пакетов. Перед тем, как удалять графический драйвер и переустановить X, нужно остановить рабочий стол. Для этого перейдите в консоль и закройте рабочий стол, сначала выйдя из него обычным образом, чтобы гарантировать, что все отключилось корректно. Когда снова появится графическое окно входа в систему, нажмите Ctrl+Alt+F1 для переключения на виртуальную консоль и войдите в нее. Затем остановите рабочий стол командой

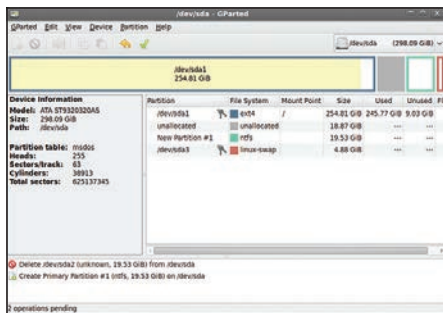
```
sudo service lightdm stop
```

Теперь удалите все следы драйвера `fglrx` командами:

```
sudo apt-get remove --purge xorg-driverfglrx fglrx*
sudo apt-get install --reinstall libgl1-mesa-glx libgl1-mesa-dri xserver-xorg-video-ati xserver-xorg-video-radeon xserver-xorg-core
sudo dpkg-reconfigure xserver-xorg
```

Эти команды удаляют все файлы проприетарного драйвера и устанавливают правильные открытые драйверы. Теперь перезагрузитесь и выполните следующую команду:

```
dmesg | egrep 'drm|radeon'
```

➤ С GParted удобно разбивать диски, но большинство установщиков дистрибутивов избавят вас от этой необходимости.

Она должна показать, что драйвер Radeon (это название открытого драйвера) загружен. С открытым драйвером Вы сможете избежать таких проблем после обновлений, так как разработчики дистрибутива уж обязательно заставят все компоненты дистрибутива работать вместе, а с проприетарными программами это не гарантируется.

7 Новый Wi-Fi, старый дистрибутив

В У меня ноутбук Toshiba Satellite с Ubuntu 12.04. На нем не поддерживается мой драйвер Wi-Fi, rtl8723ae. Не могли бы вы мне посоветовать, как лучше всего установить драйвер, или сообщить, поддерживается ли этот драйвер в 13.04?

Рэй Хьюз [Ray Hughes]

Это относительно новый дистрибутив, а на то, чтобы драйверы добрались до дистрибутива, требуется некоторое время. Во-первых, нужно получить информацию о драйвере, затем нужно добавить драйверы в последнее тестируемое ядро, после чего драйверы должны пройти через стабильные релизы ядра, и после этого еще придется подождать, пока выйдет версия дистрибутива как минимум с этой версией ядра.

Первой стабильной веткой ядра с драйвером rtl8723ae была 3.8, и она используется в Ubuntu 13.04. Можно загрузить драйвер с сайта Realtek, скомпилировать его самостоятельно и надеяться, что он заработает (отдельный драйвер достаточно привередливо относится к версиям ядра) или попробовать NDISwrapper, чтобы воспользоваться драйвером Windows.

Также можно попробовать установить ядро серии 3.8 (или более позднее) в Ubuntu 12.04, но пока самое простое решение — обновиться до 13.04. Переустанавливать систему для этого не потребуется, достаточно воспользоваться опцией `dist-upgrade` — либо в центре управления программами, либо открыв терминал и выполнив следующие команды:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

Когда Вы будете это читать, уже выйдет или будет на подходе Ubuntu 13.10, и Вы можете переключиться сразу на более позднюю версию.

Если драйвер уже появился в ядре, удаляется он оттуда редко, поэтому Ваша карта Wi-Fi должна

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 — *lshw* (<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
```

поддерживаться всеми релизами ядра и, следовательно, почти всеми дистрибутивами в обозримом будущем. **LXF**



Часто задаваемые вопросы

KVM

➤ **KVM** — это не коробочка для подключения одной клавиатуры и мышки к двум компьютерам?

Есть такая, но мы про другой KVM. Этот KVM [Kernel-based Virtual Machine] — виртуальная машина на основе ядра.

➤ **Ого! С первым вроде ясно, а зачем нужен второй?**

Он позволяет запускать виртуальные машины эффективнее (или быстрее) с помощью расширений виртуализации, встроенных в последние модели процессоров Intel и AMD.

➤ **То есть это относится к устройствам, а не к Linux?**

Отчасти. Расширения встраиваются в устройства, но в ядре находятся ПО, которое ими пользуется. Раньше для этого использовался от-

дельный набор модулей, но сейчас все это встроено в ядро.

➤ **Как узнать, поддерживают ли KVM мой процессор и ядро?**

Запустите `cat/proc/cpuinfo` в терминале. Если в разделе с флагами есть `svm` (для AMD) или `vmx` (для Intel), то процессор их поддерживает. Если у вас относительно свежий дистрибутив, то KVM в нем почти наверняка есть. А чтобы убедиться в этом, выполните команду

```
sudo modprobe -l kvm*
```

Если получится какой-то результат, поддержка KVM включена в ядро.

➤ **Итак, у меня есть KVM в процессоре и в ядре. Что нужно сделать, чтобы повысить производительность виртуальных машин?**

Да почти ничего. *VMware* и *VirtualBox* могут использовать возможности KVM автоматически, хотя нужно убедиться, что загружен модуль

`kvm-intel` или `kvm-amd` (добавьте его в список модулей, загружаемых во время загрузки системы).

➤ **Кто-то еще пользуется KVM?**

Есть версия *Qemu* специально для работы с KVM. В обычной версии *Qemu* процессор и прочие устройства компьютера эмулируются, и она может сильно тормозить, а версия с KVM работает очень быстро. В ней нет прибамбасов *VMware* и *VirtualBox*, но исходный код ее открыт, она подлинно свободна.

➤ **Это звучит несколько враждебно, а я к VirtualBox привык. Это плохо?**

Qemu работает с командной строки (сама, а не запускает виртуальные машины), но существует удобный графический менеджер для виртуальных машин KVM, который называется — внимание! — *Virtual Machine Manager* (в менеджере пакетов он скорее всего присутствует как `virt-manager`).

➤ **Все это очень хорошо, но существуют ли какие-то проблемы?**

KVM используется уже некоторое время, поэтому в целом он стабилен и надежен. Однако при попытке одновременно запустить две различные системы виртуализации, например, *VirtualBox* и *Qemu*, обе они могут попытаться использовать одни и те же расширения процессора, и это приведет к блокировке.



➤ **Virtual Machine Manager умеет создавать виртуальные машины для систем виртуализации на основе KVM и управлять ими.**



LXF HotPicks



Майк Сондерс

Издавив самые недостижимые и укромные уголки Интернета, Майк точно знает, где таятся главные сокровища открытого кода.

Thunderbird » XRoar » Mars Sim » CropGUI » DigiKam » Fgallery » Checkbashisms » Dust Racing 2D » Chrzaszcz » bd » SeaMonkey

Клиент электронной почты

Thunderbird

Версия: 24 Сайт: <http://mzl.la/15NMss2>

В конце 1990-х почти все пользователи настольными клиентами электронной почты. Были всякие предложения от Hotmail, Yahoo и прочих, но они были корявыми, медленными, и им катастрофически не хватало функций по сравнению с родными приложениями.

Появление Google Mail изменило все: пришел быстрый, привлекательный сервис webmail со множеством функций, и (благодаря использованию Ajax) он вел себя, как обычная программа рабочего стола. Сегодня большинство из нас используют webmail, однако по-прежнему остается немало причин выбрать родное приложение.

Главное преимущество родного приложения — оффлайн-поддержка. Почта

хранится на вашем компьютере, и вы имеете к ней доступ, когда потребуется, независимо от того, соединены ли вы с Интернетом. Это также удобно, если у вас мобильный тариф: хранение вашей почты локально означает, что вы не занимаете канал, читая старые сообщения и получая доступ к их вложениям.

Вы наверняка слышали о *Thunderbird*, поскольку это один из популярнейших в мире клиентов электронной почты,

«Можно просматривать или игнорировать всю цепь сообщений.»



» В *Thunderbird* также включен небольшой клиент чата с поддержкой сообщений на Facebook, Google Talk и IRC.

и им занимаются те же разработчики, что и *Firefox*. Поэтому мы здесь не будем копаться в каждой функции, а устроим вам быстрый тур по новым дополнениям и интерфейсу. Чтобы его запустить, надо распаковать файл `thunderbird-24.0.tar.bz2` и запустить *Thunderbird* в получившейся директории. Устанавливать его на всю систему не обязательно, но если вы этого хотите, распакуйте его в `/usr/lib/thunderbird/` и добавьте программу запуска или менеджер окон.

При первом запуске *Thunderbird* вам предложат ввести свой адрес электронной почты и пароль, и клиент попытается соединиться с соответствующим сервисом электронной почты. Во многих случаях вам придется выбирать между протоколами IMAP или POP3 — первый только загружает заголовки электронной почты и скачивает контент по требованию, а последний загружает все сообщение и приложения на ваш жесткий диск.

Среди основных новых функций в *Thunderbird 24*: теперь можно просматривать или игнорировать всю цепь сообщений; можно изменять масштаб в окне Compose (а не только варьировать размер шрифта); добавлена поддержка IDN-адресов электронной почты (т.е. международных адресов, использующих Unicode); и в чате IRC ники пользователей выделяются при упоминании.

Исследуем интерфейс Thunderbird

1 Вкладки

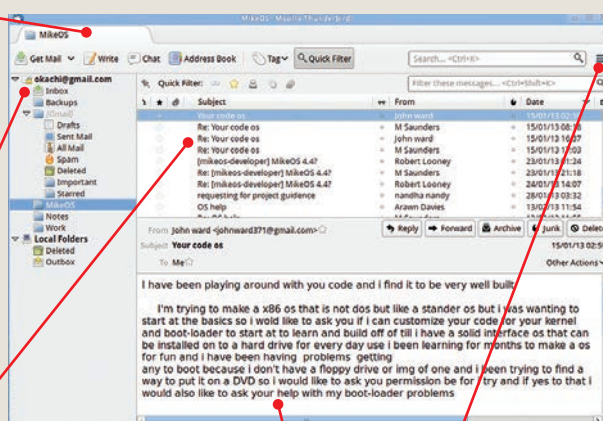
Вкладка Default показывает почту, однако вы можете открыть и другие, чтобы узнать другие функции *Thunderbird*, например, щелкнув по кнопке Chat.

2 Учетные записи почты

Здесь показываются ваши учетные записи электронной почты и соответствующие папки. Если вы вышли в учетную запись Google Mail, метки отображаются как папки.

3 Список сообщений

Непрочитанные сообщения выделяются жирным шрифтом. По умолчанию более новые сообщения появляются внизу; это можно изменить, щелкнув по заголовку колонки Date.



4 Отображение сообщения

Содержание сообщения отображается здесь. К сожалению, *Thunderbird* не умеет превращать чушь с синтаксическими ошибками в нечто более читаемое.

5 Меню

Меню *Thunderbird* не отображается по умолчанию — получите к нему доступ через кнопку с тремя линиями.

Эмулятор Dragon 32/64

XRoar

Версия: 0.30.2 Сайт: <http://bit.ly/18T0a10>

Мы с трепетом вспоминаем горячие перепалки 1980-х годов по поводу ZX Spectrum vs Commodore 64. В основном они были вполне физическими, поскольку в те времена еще не были столь развиты онлайн-коммуникации. Однако периодически кто-то вступал в дискуссию и вбрасывал в нее лихо закрученный мяч. Вы буквально размазывали очередного владельца C64, поскольку ему требовался внешний ленточный накопитель (а в вашем-то любимом Spectrum +2A был встроенный)... и тут кто-нибудь говорил: «А у меня Dragon 64, и он надерет задницу обоим вашим машинешкам».

The Dragon 32/64 (цифры относятся к встроенной памяти, в К) были не слишком широко известными 8-битными компьютерами, созданными в начале 1980-х. С точки зрения спецификаций они не представляли особого интереса — 6809 CPU, работающий на 0,89 МГц и Microsoft BASIC на ПЗУ — но они производились в Уэльсе, отсюда и название [на флаге Уэльса —

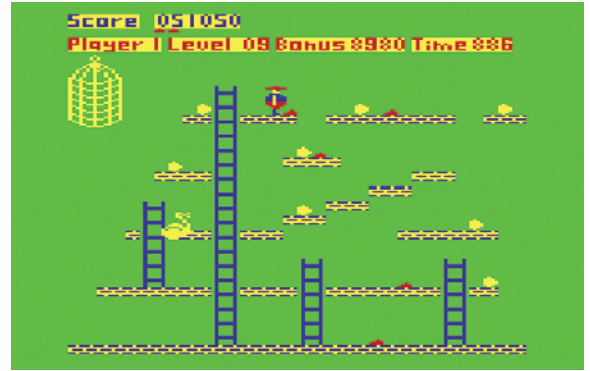
дракон, — прим. ред.]. Несмотря на то, что их выпуск прекращен в 1984 году, у этой машины в Сети до сих пор есть компания поклонников и эмулятор — XRoar.

Чтобы скомпилировать его из исходника в дистрибутивах на базе Debian/Ubuntu, установите следующие зависимости:

```
sudo apt-get install build-essential
libsndfile1-dev libgtk2.0-dev
libgtkglext1-dev libasound2-dev
```

В других дистрибутивах они доступны под теми же названиями. Затем запустите **./configure**, **make** и **make install** (последний шаг — от имени root). Если на последней стадии приключится проблема, связанная с отсутствием библиотеки, отредактируйте **config.mak**, изменив строку **LDFLAGS**, чтобы она выглядела так:

«У Dragon есть компания поклонников и эмулятор — XRoar.»



Хотя игровую библиотеку Dragon не назвать колоссальной, там есть весьма заметные названия: например, *Chuckie Egg*.

LDFLAGS = -lm

Теперь запустите процедуру компиляции снова. Прежде чем вы сможете использовать эмулятор, вам понадобится образ ПЗУ для Dragon, который надо поместить в **/usr/local/share/xroar/**; из-за потенциальных проблем с лицензией мы не вправе сообщить вам его точное местонахождение в Сети. Затем наберите **xroar**, чтобы его запустить, опционально с помощью **-default-machine**, чтобы указать тип Dragon:

xroar -default-machine dragon32

Теперь жмите **Ctrl+Shift+L**, чтобы загрузить ленточный образ (CAS, WAV или BAS) или образ диска (DMK или VDK), и сделайте автозапуск. Полный список горячих клавиш найдется в руководстве.

Планировщик высадки на Марс

Mars Simulation

Версия: 3.05 Сайт: <http://mars-sim.sourceforge.net>

Увидим ли мы колонию землян на Марсе в ближайшие десятилетия, остается догадками, но планы уже строятся. Проблема в том, что эта задача невероятно трудна. Многие кабинетные астронавты полагают, что можно просто переиграть миссию Curiosity, заменить планетоход парочкой людей — и дело в шляпе: человек на Красной планете! Однако логистика совершенно иная, нагрузка будет намного больше, и как только люди начнут разгуливать по Марсу, что они будут делать? Им нужны жилища, возможность выращивать еду, и т.д.

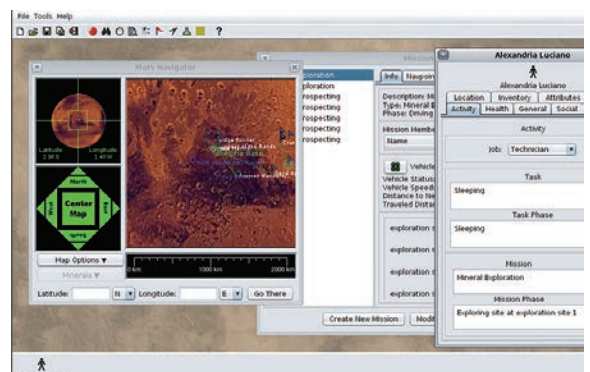
Mars Simulation Project является подробнейше детализованным приложением Java, которое подчеркивает сложности миссии на Марсе. Вы можете скачать его в виде ZIP-файла или tar-архива (после распаковки запустите **java -jar mars-simmain-3.05.jar**), и есть также Deb для Ubuntu и дистрибутивов на базе Debian, который в большинстве рабочих столов

и менеджеров окон поместит программу запуска приложения в меню Education.

Когда вы впервые запустите симулятор, вам будет предложено выбрать, каким из двух поселений вы намерены управлять: в первом население равно 24 человекам, и оно находится на довольно продвинутой стадии, а во втором — только четверо, и над ним придется потрудиться. Выберите одно из них и щелкните по **Create New Simulation** [Создать новую модель]. Появится окно руководства, и затем вы приметесь исследовать поселение.

Моделирование работает в ускоренном режиме (10 минут в поселении проходят за 1 секунду реального времени); это можно изменить в меню **Tools**, которое также

«Приложение подчеркивает сложности миссии на Марсе.»



Панель Navigator позволяет увидеть, где бродят по Марсу ваши планетоходы.

обеспечивает доступ к разнообразным видам деятельности. Вы, например, можете рассматривать разные исследовательские миссии на планете или проверять здоровье людей. Большая часть деятельности проходит без вашего непосредственного участия, однако вы можете вносить важные изменения, например, изменять расписание поставок и создавать новые задачи — скажем, постройку зданий или поиск минералов.

Самое главное, вам действительно сумеют вдолбить, какое тщательное требуется планирование, если мы надеемся когда-нибудь основать колонию на Марсе.

Программа для обрезки JPEG без потерь

CropGUI

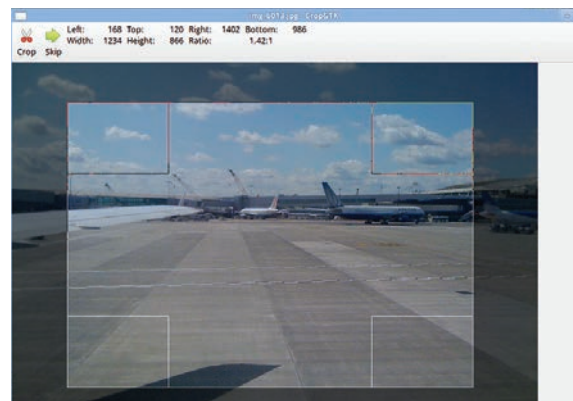
Версия: 0.1.1 Сайт: <http://bit.ly/1a5fLHW>

Вот пункт № 573 в нашем растущем списке любимых больных мозолей в Интернет: люди, применяющие формат PNG для фотографий. Ну ладно, не все знают разницу между форматами изображений, но слишком часто приходится видеть огромные фотографии, которые можно сохранить в паре МБ в формате JPEG и которые представлены в виде мастодонтов-PNG поболее 10 МБ. Большинство читателей LXF знают, какой формат выбрать, но приходится упорно твердить: JPEG идеально подходит для фото благодаря сжатию с потерями. Они могут утратить фрагменты, едва различимые глазом, чтобы уменьшить размер файла. PNG сохраняют каждый пиксель, не теряя ничего при сжатии, и потому намного больше подходят для диаграмм и графиков, логотипов и произведений цифрового искусства.

Ладно, хватить пылить. Если у вас есть несколько изображений JPEG и вы хотите вырезать из них определенные куски, то вы открываете их в GIMP, обрезаете

и сохраняете их заново. Однако при каждом новом сохранении вы должны указать уровень потерь при сжатии, то есть ранее сжатое изображение снова сжимается. Со временем это приводит к тому, что изображение размывается и появляются артефакты — примерно как копирование одной кассеты на другую, потом этой кассеты на следующую, и так далее. В конечном итоге аудио затопят помехи.

CropGUI — небольшая программа на Python, которая умеет обрезать части изображений JPEG, не сжимая их заново, что позволит избежать этой проблемы. В tar-архив включены две версии: **croptui.py** — более старая, она использует для интерфейса Tk, а **croptgtk.py** имеет более отлаженный интерфейс на GTK. Когда вы запустите программу, вам предложат



» Чудеса! Вырезание экранного снимка, показывающего процесс вырезания, не уничтожило пространственно-временной континуум.

указать местоположение файла, выбрать его, а затем перетащить уголки прямоугольника на экране до той области, которую вы хотите сохранить.

И вот здесь-то и впору растеряться. В интерфейсе нет кнопки Save, и при щелчке по Crop [Вырезать] перед вами представит окно выбора файла (чтобы открыть новый файл). Но вы не пугайтесь, потому что ранее вырезанная картинка сохранится под другим именем: так, если открывалось **image_4734.jpg**, то вырезанный фрагмент будет сохранен как **image_4734-crop.jpg** автоматически. Как только вы это узнаете, программа покажется простой и очень быстрой в использовании.

«CropGUI умеет вырезать части JPEG, не сжимая их заново.»

Фотоменеджер

DigiKam

Версия: 3.4 Сайт: www.digikam.org

Все споры на тему KDE vs Gnome в конечном итоге сводятся к спорам о настраиваемости против простоты.

Поклонники KDE обвиняют разработчиков Gnome в отуплении интерфейса и удалении важных опций, а поклонники Gnome считают KDE кошмарным лабиринтом настроек и кнопок. И эти споры выходят за пределы рабочих столов и распространяются на многие приложения, такие, как DigiKam (KDE) vs Shotwell (Gnome). Цитируем Адама Оксфорда (Сравнение, LXF168): «Инструменты редактирования Shotwell чересчур примитивны для большинства, а DigiKam может показаться перебором».

При всей своей сложности, DigiKam — очень популярный фотоменеджер. Здесь мы сконцентрируемся на его установке и рассмотрим новые функции в 3.4. Если в вашем дистрибутиве нет паке-

тов, вам придется скомпилировать его из исходника, что не так уж проблематично, если имеются нужные зависимости (*libqt*, *kdelibs*, *libkipi*, *libkdcraw*, *libkexiv2*, *libgphoto2*, *liblcms*, *libjasper*, *libtiff* и *libpng*). Загляните в файл **Readme**, где вы увидите, какие версии требуются.

Когда все будет подготовлено, скомпилируйте и установите DigiKam командами `make` и `sudo make install`

Можете добавить `-DCMAKE_INSTALL_PREFIX` в команду `make`, чтобы указать другое место установки (по умолчанию это `/usr/local/`).

«Поддержка файлов, сгенерированных Nikon Capture NX.»



» В версии 3.4 теперь возможно выбирать отдельные альбомы и теги в инструменте Maintenance Tool.

Заметнейшая новость версии 3.4 — улучшение поддержки файлов, сгенерированных редактором фото Nikon Capture NX: DigiKam стал читать их информацию Color Label. В Maintenance Tool есть новые опции для выбора альбомов и тегов для обработки, а производительность увеличена благодаря многопоточности и поддержке многоядерных CPU в инструментах Metadata Synchroniser, Thumbs Generator и Fingerprint Generator. Среди более скромных изменений — разного рода настройки по умолчанию в инструменте подавления шума, исправления в функции Face Recognition и мелкие отладки ошибок и кода.

Генератор фотогалереи

Fgallery

Версия: 1.0 Сайт: <http://bit.ly/18pukc1>

Да что же это делается с разумным дизайном пользовательского интерфейса? Последние несколько лет творится какая-то ерунда. Gnome выкинул десятки отлично отлаженных концепций ради самой большой глупости, которую можно себе вообразить, а Microsoft решил, что всем нужен интерфейс планшета, включая владельцев 32-дюймового монитора. Даже самая последняя Apple iOS напоминает явление из Мобиландии.

Поэтому, открыв *Fgallery*, мы пришли в телячий восторг. Это программа Perl, которая генерирует сетевые фотогалереи из ваших коллекций изображений, и интерфейс для этих галерей имеет отличный баланс простоты, функциональности и стиля. Он уделяет практически все внимание навигации и показу изображений, отображая лишь выжимки информации.

Если вы работаете в Debian/Ubuntu, найдите нужные зависимости таким образом:

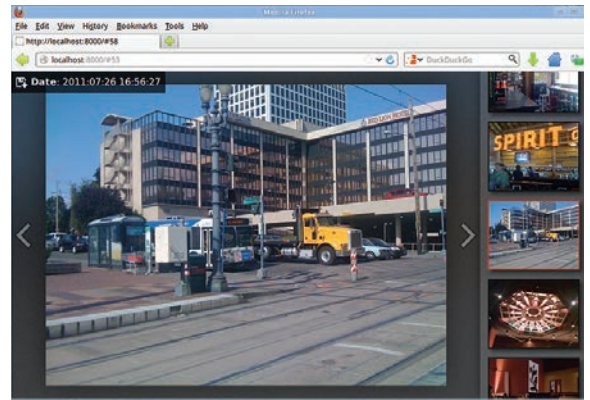
```
sudo apt-get install imagemagick
exiftran zip libjson-perl libjson-xs-perl
libtimedate-perl
```

Эти пакеты должны быть доступны и в остальных основных дистрибутивах. Как только все будет готово, распакуйте **fgallery-1.0.zip** и скопируйте содержимое директории View в папку назначения для вашей галереи (например, **web-upload**). Затем запустите

```
./fgallery /path/to/photos /path/to/
web-upload
```

Теперь, перейдя в **/path/to/web-upload**, вы увидите **index.html** — но не пытайтесь открыть его в своем браузере. Поскольку *Fgallery* использует Ajax, который обычно не поддерживается для локально хранимых файлов в браузерах, вам где-то нужно будет сделать **/path/to/web-upload** доступной через web-сервер. Это можно быстро сделать, перейдя в директорию и введя

```
python -m SimpleHTTPServer 8000
```



► Помимо использования значков со стрелками, вы можете осуществлять навигацию по изображениям клавишами курсора.

Загляните на <http://localhost:8000>, чтобы полюбоваться галереей в своем браузере.

Fgallery автоматически меняет масштаб и сжимает изображения, чтобы они быстрее демонстрировались в Сети, а также урезает данные EXIF.

Интерфейс быстрый и простой: эскизы справа, полные изображения слева, а информация по дате отображается сверху. Вы также увидите значок с флорпи-дискон — это ссылка на ZIP-файл исходных изображений, с неизменным масштабом.

«Имеет отличный баланс простоты, функциональности и стиля.»

Проверка портатбельности скриптов

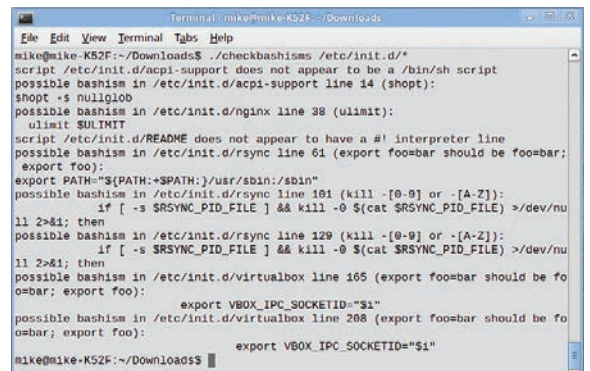
Checkbashisms

Версия: 2.0.0.2 Сайт: <http://bit.ly/19W0hLm>

Как и мы, вы, вероятно, проводите большую часть своего времени в Linux, совершая периодические вылазки в другие ОС с открытым кодом — или Unix-подобные: FreeBSD, NetBSD, возможно, даже немного Solaris или AIX. А если вы читаете форумы и новостные группы этих ОС, вы знаете о муссирующей там теме: пользователи жалуются, что слишком много программ с открытым кодом ориентировано на Linux. Вместо того, чтобы программировать по нейтральным к ОС стандартам, таким, как POSIX, многие разработчики предполагают, что все пользуются Linux, и добавляют в свой код сугубо линуксовые части.

Хороший пример — скрипты оболочки. *Bash* во многом является фактическим стандартом Linux, но вы не можете гарантировать этого во всех дистрибутивах, и в других Unix-подобных ОС оболочкой по умолчанию (**/bin/sh**), возможно, будет

не *Bash* (даже если *Bash* доступен или установлен везде). И если вы пишете скрипт, желая, чтобы он работал везде, избегайте ориентированного на *Bash* кода. Это легче сказать, чем сделать, однако *Checkbashisms* вам поможет. Этот скрипт Perl проникает внутрь скриптов *Bash* в поисках потенциальных проблем с совместимостью с простой оболочкой **/bin/sh Bourne**. Чтобы он заработал, скачайте его и сделайте исполняемым (**chmod +x checkbashisms**). Затем можете запустить его на месте с одним или более скриптами оболочки в качестве параметра — скажем, чтобы увидеть, насколько ориентированы на *Bash* скрипты загрузки вашего дистрибутива, введите



► Не будем суровы к Ubuntu за ориентированные на *Bash* скрипты, поскольку там это имеет смысл, но данный пример показывает возможные проблемы.

```
./checkbashisms /etc/init.d/*
```

Тут *Checkbashisms* покажет потенциально проблемные строки, а часто подскажет и альтернативы, по типу `export foo=bar` должно быть `foo=bar; export foo`

В данном случае это проблема синтаксиса, и вы также увидите предупреждение о встроенных командах, которые не предоставляются стандартной **/bin/sh** реализацией: например, **shopt** и **alias**.

В итоге исправления этих проблем ваш код скорее всего удлинится и усложнится, вызвав у вас ощущение, что вы применяете более примитивную оболочку; однако такова цена лучшей переносимости.

«Checkbashisms покажет потенциально проблемные строки.»

HotGames Развлекательные приложения

Гонки

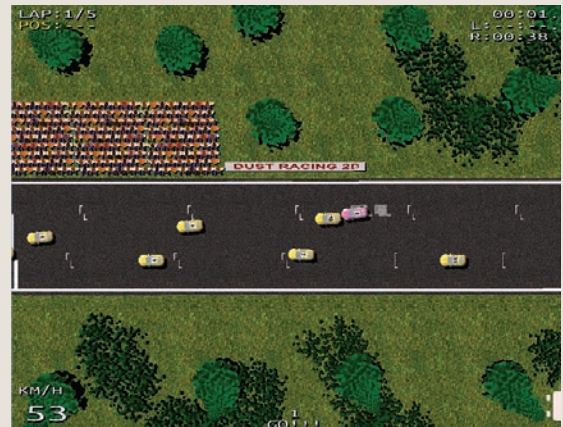
Dust Racing 2D

Версия: 1.4.3 Сайт: <http://bit.ly/17xKRqT>

Читатели LXF старшего поколения, возможно, помнят *Super Sprint*. Эта инновационная гонка-аркада предлагала наличие руля и поддерживала одновременно трех игроков. Игра была проста: вам давали статичное отображение гоночной трассы сверху вниз, и вашей целью было пройти ее, победить соперников и выиграть три круга.

Dust Racing 2D во многом вышла из *Super Sprint*, но наворотов тут куда больше: есть прокрутка трассы и детальная прорисовка. Очень удобно, что не надо компилировать ее из исходника: помимо пакетов для дистрибутивов на базе Debian/Ubuntu, есть прекомпилированные двоичные tar-архивы. Скачайте [dustrac-1.4.3-linuxi686-qt4.tar.gz](http://bit.ly/17xKRqT) (или альтернативу `x86_64`, если у вас 64-битный дистрибутив), распакуйте его и запустите `./dustracgame` в полученной директории.

Нажмите на Enter в меню Play, чтобы начать игру. Изначально у вас есть доступ только к одной трассе, но по мере того, как вы будете выигрывать гонки, трасс станет больше. В начале каждой гонки ваша розовая машинка находится в конце сетки, позади 11 своих управляемых CPU соперников — и те настроены явно не по-детски. Используйте клавишу стрелки вверх, чтобы разогнаться, стрелки вниз, чтобы тормозить, и стрелок влево и вправо для управления. Вначале игра может показаться кошмарно сложной, поскольку машинка проскальзывает, но немного спустя



► Фоновый ландшафт в основном весьма статичен, кроме некоторого качания деревьев.

вы достигнете нужного уровня мышечной реакции, чтобы вписываться в повороты, не слетая с трассы.

Чтобы перейти на следующую трассу, вы должны войти в число шестерых финалистов. В игре только шесть трасс (парочка напоминает трассы Формулы-1), но вы можете добавить собственные через имеющийся редактор уровней. Однако не пытайтесь настраивать аудио, не услышав в игре звуков — на сей раз это не PulseAudio снова барахлит. Просто *Dust Racing 2D* — игра тихая до жути.

«Dust Racing 2D вышла из Super Sprint, но наворотов тут больше.»

Лабиринт

Chrzazscz

Версия: 0.1.0 Сайт: <http://bit.ly/H70kpb>

Если вы поляк, поздравляю вас: возможно, вам удастся это выговорить. Пожалуйста, сделайте нам одолжение: зайдите на www.uscagoo.com, запишите свое произношение этого названия и поделитесь результатом на наших форумах. Так или иначе, *Chrzazscz* — игра-исследование лабиринта; она находится на ранней стадии разработки и создана на библиотеках SDL, а чтобы скомпилировать ее из исходника, понадобятся также *SDL-mixer*, *SDL-image* и *SDL-ttf*.

После терзающего гитару вступления на экране ввода, *Chrzazscz*, подобно *Dust Racing 2D*, проходит в полной тишине. Нажмите на Enter, чтобы выбрать уровень игры — здесь их пять, и ко всем есть доступ с самого начала. Затем вам предложат выбрать, за какого из персонажей вы будете играть (воина, вора или волшебника), и вы попадете в лабиринт.

Сперва вы видите только одну комнату, но когда начнете проходить в разные двери, видимая часть карты станет больше.

Здесь нет схваток или магических заклятий; это исследование. Нажатие на F1 покажет список ваших целей, которые всегда вовлекают посещение определенных частей лабиринта. Некоторые из этих частей прячутся за закрытыми дверями, и чтобы получить соответствующие ключи с цветовым кодом, вам придется пообщаться с голубыми орками, снующими вокруг. Однако за раз вы можете поиметь только два ключа, а некоторые орки обменивают ключи только определенного цвета, так что вам придется планировать



► Прохождение уровня само по себе радует, но добавление трех восклицательных знаков выводит на новый уровень эйфории.

свои действия заранее. Не очень способствует успеху и то, что орки могут исчезнуть в комнатах, где вы еще не были (и которые, следовательно, невидимы)...

Chrzazscz еще на очень ранней стадии разработки, и многих элементов пока не хватает, но игра выглядит многообещающей и уже готовой к бою. Если в нее добавить уровней, чуть поработать над зрелищной стороной и чуть разнообразить типы игроков (сейчас разница между ними чисто косметическая), однажды она может стать классикой.

«Chrzazscz выглядит многообещающей и уже готовой к игре.»

Быстрый переключатель директорий

Bd

Версия: N/A Сайт: <https://github.com/vigneshwaranr/bd>

Xотя мы любим открывать для HotPicks большие и сверкающие новые приложения, не меньше нам нравится наткнуться на чудесные программки-крошки, экономящие время. Превосходный пример — *bd*: это скрипт оболочки, действия которого не особо картинны, но за недели и месяцы он способен сэкономить вам немало времени, если вы частый посетитель командной строки.

Вот как он работает. Пусть вы находитесь в глубоко зарытой директории, например, `/usr/lib/libreoffice/program/wizards/ui/event`, и вам надо перейти на несколько уровней вверх. Чтобы попасть в `/usr/lib/libreoffice`, вам бы пришлось ввести это:

```
cd ../../../../
```

Не шибко элегантно. Да еще придется все время высчитывать, куда вы попали, для каждой вводимой пары точек. Не лучше ли было бы просто ввести отдельно имя директории более высокого уровня? Именно это и позволяет сделать *bd*:

```
bd libreoffice
```

Здесь *bd* возвращает вас в `/usr/lib/libreoffice/`, и вам не придется указывать весь путь или ставить множество точек. Если у директорий нет конфликта имен, можно даже не вводить имя полностью: **bd libre** окажет тот же эффект. Для тех, кто проводит много времени в глубине директорий (что вполне обычно для разработчиков, которые трудятся над большими кодовыми базами), он должен стать главным оружием в арсенале CLI.

```

Mike@Mike-K52F:~/usr/lib/libreoffice/program/wizards/ui/event$ ls
total 228K
-rw-r--r-- 1 root root 59K Mar 26 12:02 CREDITS.odt
-rw-r--r-- 1 root root 138K Mar 26 12:02 LICENSE.odt
-rw-r--r-- 1 root root 4.0K Mar 26 12:02 NOTICE
drwxr-xr-x 15 root root 4.0K May 22 16:54 presets
drwxr-xr-x 10 root root 12K May 22 11:03 program
drwxr-xr-x 20 root root 4.0K May 22 11:03 share
lrwxrwxrwx 1 root root 6 Apr 11 21:25 ../share -> ../share
Mike@Mike-K52F:~/usr/lib/libreoffice/program/wizards/ui/event$ ls
total 228K
-rw-r--r-- 1 root root 59K Mar 26 12:02 CREDITS.odt
-rw-r--r-- 1 root root 138K Mar 26 12:02 LICENSE.odt
-rw-r--r-- 1 root root 4.0K Mar 26 12:02 NOTICE
drwxr-xr-x 15 root root 4.0K May 22 16:54 presets
drwxr-xr-x 10 root root 12K May 22 11:03 program
drwxr-xr-x 20 root root 4.0K May 22 11:03 share
lrwxrwxrwx 1 root root 6 Apr 11 21:25 ../share -> ../share

```

Используйте обратный апостроф, чтобы вывести список файлов в директориях более высоких уровней, например: `ls -l `bd libre``.

Интернет-пакет

SeaMonkey

Версия: 2.21 Сайт: www.seamonkey-project.org

Oн не умер! *SeaMonkey* в наши дни является невероятно незаметным проектом, несмотря на всю свою завораживающую историю. Для тех, кто с ним не встречался: *SeaMonkey* — наследник пакета Mozilla, которая, в свою очередь, наследовала *Netscape*, браузеру, который господствовал на домашних компьютерах в конце 1990-х (пока Microsoft не вбил всем в глотку *Internet Explorer*).

Сегодня популярность отвоевали две отдельные программы, являющиеся ответвлениями кодовой базы Mozilla — *Firefox* и *Thunderbird*; однако без особой шумихи ведется и разработка *SeaMonkey*. Проект сочетает разные интернет-инструменты в одном приложении, что некоторые могут назвать раздутостью и отходом от философии Unix; но некоторые считают очень удобным иметь все в одной посуде: *SeaMonkey* сочетает в себе web-браузер, клиент электронной почты, программу IRC-чата и редактор HTML. Раскладка GUI

не слишком отличается от того, что было в дни Mozilla и Netscape, и если вы скучаете по браузеру, который не стремится все упростить донельзя и избавиться от кнопок во имя «простоты», дайте ему шанс.

Скачав **seamonkey-2.21.tar.bz2**, распакуйте его — и запускайте **seamonkey** в полученной директории. Нет нужды устанавливать его в систему, и там, где это применимо, программа возьмет ваши настройки *Thunderbird*. При первом запуске вам также предложат сделать *SeaMonkey* браузером по умолчанию. **LXF**



SeaMonkey показывает экранный снимок семилетней давности пакета Mozilla — заметьте, раскладка GUI практически та же.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

Beets 1.3.0

Усовершенствуйте свою музыкальную коллекцию лучшими метаданными. beets.radbox.org

deheader 0.8

Избегайте включения файлов с дублированными заголовками в свой исходный код C/C++. www.catb.org/~esr/deheader/

BirdFont 0.30

Редактор шрифтов, способный создавать шрифты TTF, EOT и SVG. <http://birdfont.org>



Загляните в LXF175 HotPicks, чтобы узнать об этом приложении подробнее.

IceWM 1.3.7

Классический менеджер окон, который вовсе не так уж мертв, как думали некоторые. www.icewm.org

Gnome Chemistry Utils 0.14.2

Пакет с химическим редактором, калькулятором и просмотрщиком молекул 3D. <http://gchemutils.nongnu.org>

Emacs 24.3

Предмет страстной любви и жгучей ненависти — редактор GNU, теперь с системой пакетов. www.gnu.org/software/emacs/

F-IRC 1.10

Основанный на консоли клиент IRC, стремящийся к простоте в использовании и навигации. www.vanheusden.com/fi/

Rescue! Max

Космическая игра в жанре экшн-приключение, написанная на Java. <http://rescue.sourceforge.net>



Правда, все выглядит весьма ...образно?

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Загрузка

Выбор — это хорошо; это одна из присказок мира Linux. Значит, и изменения — хорошо, да? Иначе откуда взяться выбору! Но изменение ради изменения... Где смысл?

Мы поэкспериментировали с применением *Grub* для загрузки LXF DVD, потому что в нем есть милая функция: загружать директорию с ISO-образа, хранящегося на DVD, так что вам придется загрузить дистрибутив с DVD или записать собственный диск с файла ISO. Но системы загрузки дистрибутивов должны это поддерживать; и многие поддерживают.

Это делается по-разному, в том числе используя аргументы передачи с командой загрузки. У разных дистрибутивов разные методы, но все они сводятся к одному. Метод *System Rescue CD* работает хорошо, а процедура *Ubuntu* работает не хуже. И так, есть изменение и есть выбор, и все ОК.

Но зачем дистрибутиву на базе *Ubuntu* сливать готовый, рабочий метод, сменив его на нечто недокументированное и не срабатывающее даже после перебора всех опций? Дистрибутив называть не буду, и его нет на DVD этого месяца. Разработчики, пожалуйста, делайте как лучше для нас или проще для вас, а не потому, что руки чешутся.

Neil

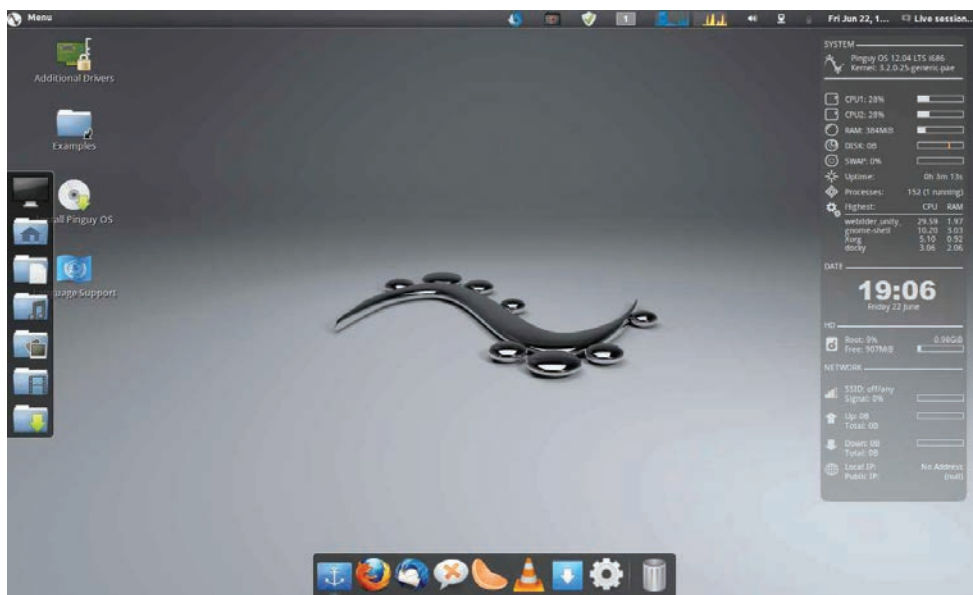
» Важно ВНИМАНИЕ!

Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

Дистрибутив Linux

PinguyOS



В сравнении этого месяца у нас были дистрибутивы для начинающих пользователей, и вполне логично ввести парочку кандидатов в состав DVD. Всех включить не удалось, поскольку некоторые очень объемны, и нам бы пришлось перейти на диски Blu-Ray. *PinguyOS* хорошо приняли

в качестве производного *Ubuntu*, с которым легко освоиться, и вот он перед вами. Он намного больше стандартного *Ubuntu CD*, а значит, вы получите готовую операционную систему, причем со всеми нужными программами, так что дополнительные пакеты искать не придется.

Дистрибутив Linux

Elementary OS

Другой «дружелюбный к начинающим» дистрибутив на DVD — *Elementary OS*, рабочий стол которого покажется знакомым пользователям *Mac OS X*. Мы понимаем, что не все наши пользователи — начинающие линуксоиды (иначе стоило бы подивиться, кто до этого года покупал наш журнал), но дистрибутив, хороший для новичков, может быть не хуже и для всех остальных.

В конце концов, кто ж не полюбит дистрибутив, который легко устанавливается, определяет оборудование и позволяет делать все, что хочется, без лишних трудов? Если ваш ответ на этот вопрос — «я», и если вам нужно нечто хардкорное, которое по умолчанию загружается в консоль, то у нас есть *System Rescue CD*. Это один из лучших live CD для восстановления

системы, он идет с 32- и 64-битными ядрами, и в нем множество инструментов для восстановления или настройки систем Linux и Windows.





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

Сравнение

Знакомьтесь со звездами среди дистрибутивов для новичков.

Учебники

ПО для учебника про личный рабочий стол и многое другое.

ПО для сервера

Советы и программы, чтобы достичь Высокой Отказоустойчивости.



LXFHotPicks

Checkbashisms

Проводит базовую проверку скриптов оболочки на предмет наличия непортируемого синтаксиса.

DigiKam

Тем, кто ищет продвинутое приложение для управления цифровыми фотографиями для KDE.

DustRacing2D

Мозаичные кроссплатформенные 2D-гонки, написанные на Qt (C++) и OpenGL.

Chrzazscz

Лабиринт с открытым кодом... в котором суетятся голубые (не по ориентации, а по окраске) Орки!

Fgallery

Генератор статической фотогалереи — без наворотов, но с минималистски-стильной внешностью.

Mars Simulation

Проект Java, относящийся к свободному ПО, для моделирования будущего человеческого поселения на Марсе.

Thunderbird

Последняя версия приложения электронной почты, которое легко настраивается и предлагает отличные функции.

SeaMonkey

Интернет-пакет все-в-одном.

Bd

Быстро возвращает вас в родительскую директорию в оболочке.

CropGUI

Программа на GTK для обрезки изображений JPEG без потерь.

XRoar

Эмулятор антикварного компьютера Dragon 32/64.



И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязательные присутствовать в каждой операционной системе.

Hardinfo Инструмент для тестирования системы.

Plop Простой менеджер загрузки для запуска операционных систем.

RaWrite Создавайте загрузочные диски в Windows.

SBM Независимый от ОС менеджер загрузки с простым в использовании 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.

Linux Kernel in a Nutshell [Ядро Linux в двух словах] Введение в ядро, написанное великим магистром Греггом Кроа-Хартманом [Greg Kroah-Hartman].

Справочник администратора Debian Руководство для системных администраторов.

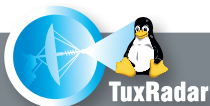
Словарь Linux Linux от A до Z.

Окунитесь в Python Мастер-класс по этому популярному языку.

Руководство по созданию скриптов Bourne Shell Начните осваивать скрипты оболочки.



Подкасты



S5 E17 Как в легенде о фениксе

Новости [News] Барри Каулер [Barry Kauler], основатель Puppy Linux, ушел в отставку (опять), Valve анонсирует основанную на Linux SteamOS для своих Steam Machines, Nvidia помогает драйверам с открытым кодом, и тут появляется CyanogenMod.

Открытое голосование [Open Ballot] Глава Steam, или просто пар выходил?

Открытия недели [Discoveries of the week] Бен указывает дорогу в Hell, Майку нравится Winning, а Грэм, Эндрю и Эффи, похоже, в прошлые месяцы только и делали, что играли в игры (*Sir, You Are Being Hunted*, *Pixel Junk Monsters* и *Little Master*).

Предупреждение читателю: У этого подкаста печальный конец. Готовьте платочки!



Пропустили номер?



Закажите его через «Линуксцентр» по адресу www.linuxcenter.ru! 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:

LXF175

Октябрь 2013

- » Сисадмины будущего Прокладываем новые пути
- » Звуки музыки Доверьтесь Поток
- » rump.io Движок для открытых социальных сетей
- » Сайт в гуще жизни Как раскрутиться в Интернете

LXFDVD: Fedora 19 (64-разрядная сборка с Gnome 3), Arch Linux, Knoppix, Tails, Overclockix, CentOS, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_175/
PDF-версия: shop.linuxformat.ru/elxf_175/



250 руб.

LXF176

Ноябрь 2013

- » Строим свой дистрибутив Даже со своим логотипом
- » Скучным рыцарям Коллекция под контролем
- » Open vSwitch Виртуальным серверам — единоначалие
- » Я тучка, тучка, тучка... Облака нынче в моде

LXFDVD: ROSA Fresh R1 (64-разрядная сборка с KDE), Peppermint 4, Sonar 13.10, Ubuntu 13.10, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_176/
PDF-версия: shop.linuxformat.ru/elxf_176/



250 руб.

LXF177

Декабрь 2013

- » Дрожи, Android Конкуренты давно уж подпирают
- » Держите в курсе Собираем сплетни читалками новостей
- » Rust Язык от взломщиков защитит
- » Кстати о взломщиках LMD на страже системы

LXFDVD: openSUSE 13.1 (64-разрядная сборка), Grml 2013.09, LXLE, OS/4 OpenLinux, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_177/
PDF-версия: shop.linuxformat.ru/elxf_177/



250 руб.

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! Через shop.linuxformat.ru вы также сможете приобрести предыдущие выпуски LXF (если успеете).

Подписывайтесь на 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
журнала Linux Format
ВКонтакте вы найдете:

- » Множество новостей 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: X2Go, Bluez 5, Perl 5.18 и другие обновления

Fedora 20



Ядро 3.11 » Gnome 3.10 » Gimp 2.8 » KDE Plasma Workspaces 4.11

64-разрядная сборка

Январь 2014
LXF DVD 178/179

LINUX
FORMAT

Январь 2014
LXF DVD 178/179

LINUX
FORMAT

PinguOS 12.04

LTS-релиз, 64-разрядная сборка

Еще дистрибутивы: Elementary OS Lina » SystemRescueCD 3.8.1

А ТАКЖЕ: Thunderbird 24, Digikam 3.4, 10 книг о Linux и многое другое



Сторона 1

ДИСТРИБУТИВЫ

Elementary OS Luna Дистрибутив для начинающих пользователей с рабочим столом в стиле Mac OS, 32- и 64-разрядные сборки (ISO-образы)

PinguOS 12.04 Популярный легкий дистрибутив на базе Ubuntu, 64-разрядная сборка (загрузка с LXFDVD)

SystemRescueCD 3.8.1 Live CD с набором средств для восстановления системы. Комбинированная 32- и 64-разрядная сборка (ISO-образ)

Сторона 2

ДИСТРИБУТИВЫ

Fedora 20 Обновленная версия популярного дистрибутива. Установочный DVD, 64-разрядная сборка (загрузка с LXFDVD)

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

hd Скрипт оболочки для быстрого переключения текущей директории

Checksums 2.0.0.2 Программа проверки портательности скриптов

Chrtaszcz 0.1.0 Игра, лабиринт

StripGUI 0.1.1 Программа для обрезки JPEG без потерь

DigiKam 3.4 Менеджер фотографий

Dust Racing 2D 1.4.3 Игра, автомобильные гонки

Fgally 1.0 Генератор фотогалереи

Mars Simulation 3.4 Планировщик высадки на Марс

SeaMonkey 2.21 Интернет-пакет, включающий браузер, редактор HTML, почтовый и IRC-клиенты

Thunderbird 24 Клиент электронной почты

XRoar 0.30.2 Эмулятор Dragon 32/64

Пожалуйста, перестаньте использовать Аджноу диск с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@lxfmagazine.com

ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефекта на данном диске, обращайтесь, пожалуйста, по адресу feedback@lxfmagazine.com

Настоящий диск тщательно тестировался и проверялся на всех старых платформах, 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 в вашем жестком диске. 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 нами программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с нашего сайта: www.linuxformat.com

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 нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *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 LINUX ТЕПЕРЬ НА USB FLASH



Linux Mint 15



Ubuntu 13.04

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/



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 ООО «Ланинь»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 8565

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@futurenet.com

Заместитель редактора Мэтью Хансон [Matthew Hanson]

matthew.hanson@futurenet.com

Научный редактор Бен Эверард [Ben Everard] ben_everard@futurenet.com

Выпускающий редактор Крис Торнетт [Chris Thornett]

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

[Elrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Джон Бэкон [John Bacon], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Марко Фиоретти [Marco Fioretti], Мэтт Хансон [Matt Hanson], Джон Лэйн [John Lane], Грэм Моррисон [Graham Morrison], Лес Паундер [Les Pounder], Джонатан Робертс [Jonathan Roberts], Майк Сондерс [Mike Saunders], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Гэри Уокер [Gary Walker], Евгений Балдин, Артем Зорин, Андрей Прахов, Алексей Федорчук, Игорь Штомпель

Художественный ассистент Донна-Мари Скразе [Donna-Marie Scrase]

Иллюстрация Шейн Коллиндж [Shane Collinge], Саймон Миддлвик [Simon Middleweek], Елси Уолтон [Elly Walton Illustrations]

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, д. 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 исключительное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

“GNU/Linux” заменяется на “Linux” в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>

В февральском номере

Mint 16 изнутри

Элегантный дистрибутив выпустил новый, 16-й релиз с мятой свежестью. Вдохнем ее во всей полноте и оценим кульминацию тяжких шестимесячных трудов.

Не только Raspberry Pi

Возведем компьютер-крошку на новый уровень, пристально рассмотрев платы расширения, которые его украсят.

Строим статический сайт

Вероятно, наискорейший способ построить и запустить собственный сайт: быстрый и отнимающий мало ресурсов.

Равноправие в Linux

Налицо заметный гендерный перекосяк. В каком направлении надо работать, чтобы привлечь в FOSS побольше женщин?

Red Hat Enterprise Linux

предоставляет вам **производительность, масштабируемость, безопасность и надежность**, ранее доступные только на очень дорогих платформах

Самая популярная в мире Linux платформа для бизнеса

Обеспечивает высокую производительность, надежность, масштабируемость и безопасность

Сертифицирована ведущими производителями оборудования и разработчиками ПО



Совместима с широким спектром оборудования от рабочих станций до серверов и мэйнфреймов

Обеспечивает одинаковые условия работы приложений при использовании в физической, виртуальной и облачной средах

Пользователи RHEL экономят на оборудовании, лицензиях на программное обеспечение и эксплуатационных расходах



ГНУ/Линуксцентр — Linux-эксперт для вашего бизнеса

- Premier Business Partner компании Red Hat
- 12 специалистов по разработке и внедрению, сертифицированных компанией Red Hat
- Более 100 клиентов, использующих Red Hat
- 10 лет на рынке

Red Hat — ведущий серверный дистрибутив Linux

- Более 15 лет промышленного использования
- Свыше 80% рынка корпоративного Linux по данным CIO Insight
- 5 лет среди лучших вендоров
- Выгодная совокупная стоимость владения (TCO)
- Поддержка в течение 10 лет

Специальное предложение!

Закажите Red Hat Enterprise Linux в ГНУ/Линуксцентре и получите в подарок книгу «Полное руководство пользователя Red Hat Enterprise Linux»



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

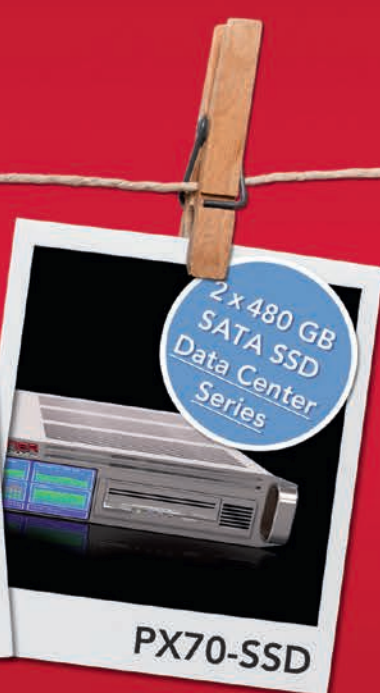
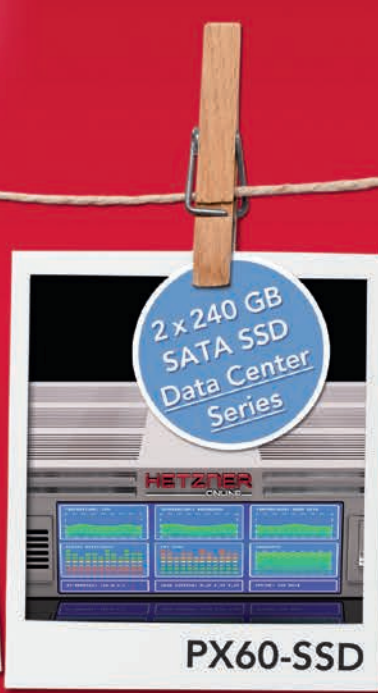
309-06-86

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. Выберите 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.