

ИЗНУТРИ: UBUNTU 14.04 LTS

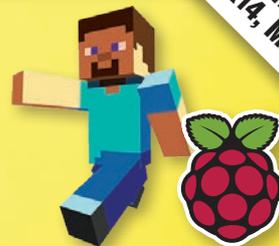
LINUX FORMAT

Главное в мире Linux

Июль 2014 № 7 (185)

СОБЕРЕМ LINUX-ПК

Рабочая лошадка,
медиа-центр
и домашний сервер —
оптимальные варианты



Minecraft на Raspberry Pi

» Изучаем основы и строим требушет (угадайте, что это) с. 86

БЕСПЛАТНЫЙ DVD
64-битный Ubuntu 14.04 LTS,
MX14, Mint 17 и сверх того!



ПЛЮС!

Что за штука —
UEFI
с. 54



Учат в школе

Камера на Raspberry Pi
записывает мои
выкрутасы на доске

Мартин О'Хэнлон о применении Pi для хобби с. 40

Также в номере...

Linux Chromebook

» Пристроим Linux
на детище Google, будет
супер-дешевый нетбук с. 50



OpenStack

Гид по DevStack

» Локальное облако
на виртуальной машине

Open Evernote

Открытые заметки

» С Etherpad и NixNote
вы ничего не забудете

3D-печать

Сработаем корпус к Pi

» Дизайн и изготовление
при помощи Python и FreeCAD

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

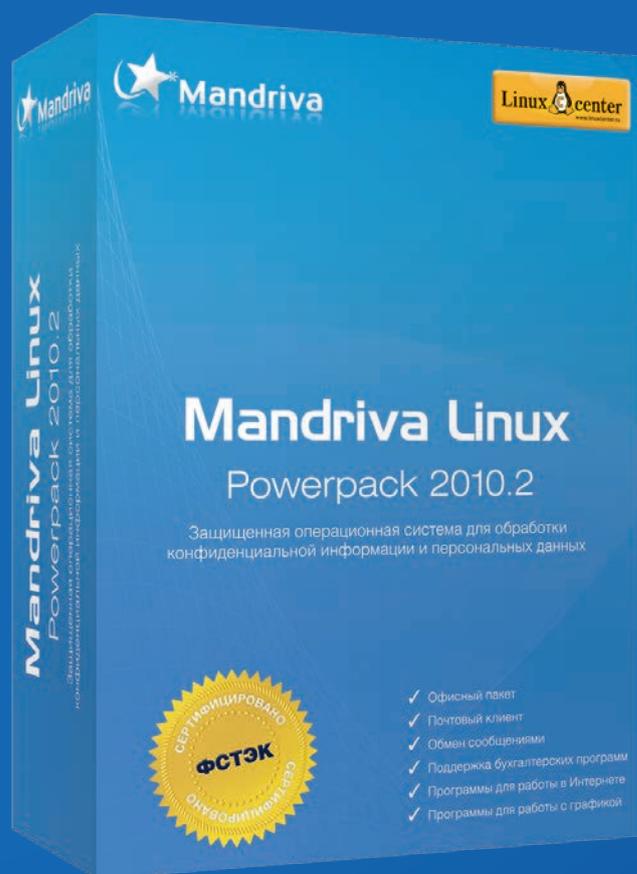
Linux center
www.linuxcenter.ru



Mandriva Linux

Powerpack 2010.2

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



Простота установки, работы и администрирования

Легкость освоения новичками в GNU/Linux

Широчайший спектр поддерживаемого оборудования

Цена 4990 руб.

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



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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Что мы делаем

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



Кто мы

В этом номере мы показываем, как собрать собственную машину с Linux. И мы спросили наших авторов: какой ПК был у них самым первым?



Джонни Бидвелл

В 1994 году один скользкий тип из Стирлинга уболтал меня сменить мою Amiga 500 на 486DX — она работала с бешеной частотой 33 МГц и обладала гигантскими 8 МБ ОЗУ. Мне нравилось возиться с `config.sys` и загружать драйвера устройств, сберегая свои драгоценные 640К.



Нейл Ботвик

Моим первым ПК был Jupiter Ace, подобие ZX81 с замечательной версией Forth. Даже тогда я применял альтернативную ОС. Естественно, для работы пришлось вставить в него самодельное ОЗУ на 16К (да-да, целых 16!). Изучать Forth было здорово интересно!



Дэвид Хейвард

Мой первый ПК — Amstrad MegaPC, гремучая смесь 386SX на 25 МГц, 1 МБ ОЗУ и 20 МБ жесткого диска с DOS и Windows 3.1; заодно был встроен Sega MegaDrive, который активировался горизонтальным сдвигом передней панели со стороны на сторону — вот такая магия.



Лес Паундер

Самый первый мой ПК — Commodore 16. Помню, как я впервые собирал оборудование и запускал систему, чтобы увидеть BASIC/командную строку. Как ребенок того поколения, я, естественно, принялся исследовать компьютер и споткнулся на коде 10 PRINT.



Маянк Шарма

Я начал программировать на GW-BASIC в 1995 году, когда учился в 7-м классе, на i486. Дома компьютер появился в 1997-м — Pentium MMX на 166 МГц, он отъел изрядный кусок отцовских сбережений. У него были 64 МБ ОЗУ, жесткий диск 1,2 Гб и даже CD-дисковод! Тогда я впервые запустил Linux.



Валентин Синицын

Scorpion ZS 256. Эх, старые добрые деньки за Dizzy... и никаких войн Vi против Emacs.



Развилка истории

» Любители альтернативной истории привыкли называть «развилкой» момент принятия некоего решения, предопределившего дальнейший ход событий. Причем зачастую в момент принятия само это решение не казалось достаточно важным.

В истории ИТ такой развилкой можно считать выбор даже не столько архитектуры, сколько конструктивного облика IBM PC. Система собиралась из «крупных блоков» — материнской платы, видео- и дискового контроллеров, объединенных системной шиной, спецификация которой была открыта для сторонних разработчиков. Возник рынок «изготовителей оригинального оборудования» (OEM's, Original Equipment Manufacturers), совместимого с новым открытым стандартом. Конкуренция между изготовителями привела к быстрому снижению цен и сегодняшнему доминированию «стандартной архитектуры». Так одно «инженерское» решение задало траекторию развития целой индустрии на четверть века вперед.

Альтернативой могла бы стать «моноблочная» система, где все компоненты монтировались бы на одной плате. И в производстве было бы дешевле, и никаких конкурентов... Только мир был бы совсем иным, а редакции LXF пришлось бы выбрать другую тему для этого номера.

Кирилл Степанов

Главный редактор

» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

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

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

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

0+

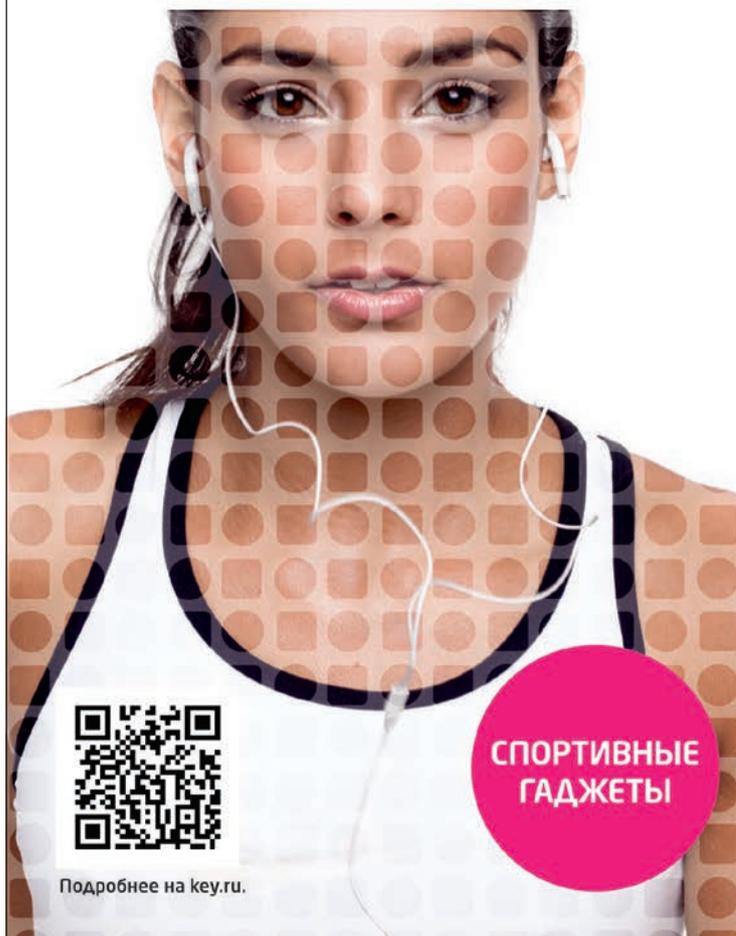


Clickme ^{hh}

Реклама своими руками

<http://clickme.hh.ru>

ТВОИ ПОМОЩНИКИ В ТРЕНИРОВКАХ



Подробнее на key.ru.

СПОРТИВНЫЕ
ГАДЖЕТЫ

Экшн - камера



Экшн-камера Sony HDR-AS30

Smart - часы



Smart-часы Cookoo Watch

Smart - здоровье



Браслет Jawbone UP



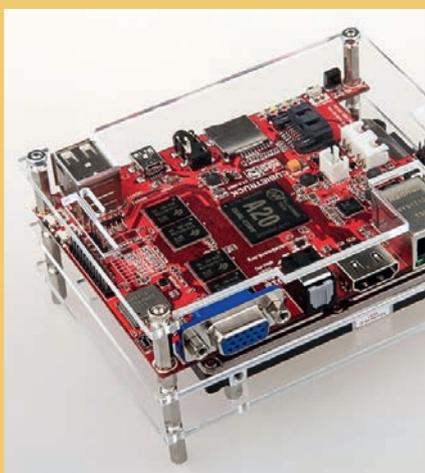
Шаромер Misfit Shine

Товар сертифицирован. Количество товара ограничено.



Одноплатный компьютер Cubietruck3

Cubietruck



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

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

литиевого элемента питания

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

КОМПЛЕКТАЦИЯ

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

4395
руб.

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

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



Содержание

Единственный журнал про Linux со встроенной защитой армейского уровня!

Обзоры

Ubuntu 14.04 LTS 14

Canonical выпустила очередную версию Ubuntu с долгосрочной поддержкой — и мы погоняли ее на корде.



► **Ubuntu 14.04 LTS: что нам предлагает данный релиз?**

LPS 1.5.1 16

Люди привыкли попусту судачить о «защите армейского уровня», но данный конкретный liveCD создан ВВС США.

Sonic Pi 17

Удовольствие от создания музыки сочетается с логикой программирования!

diskAshur Pro 18

Внешний диск с защитой армейского уровня. Шутим, шутим! Но он и вправду отлично защищен.

AOC A2472PW4T 19

Нет, это не кто-то мазнул пятерней по клавиатуре — это действительно название такое.



► **На самом деле, AOC A2472PW4T — моноблок на Android.**

Razer BlackWidow 20

Механическая клавиатура, точная и удобная в работе.

FRITZ!Box 7490 20

Один из лучших маршрутизаторов, предлагаемых рынком.

Игры 21

Мы позабавились с музыкальной головоломкой *Circuits* и полетали по космосу в *The Last Federation*.

Соберем Linux-ПК

Соберите себе ПК своей мечты дешевле 30 000 руб. с. 30



Сравнение: Платформы для википедий с. 24

The collage shows various wiki editing interfaces. One prominent screenshot is from DokuWiki, showing a 'Media Files' section with options to 'Keep window open on selection' and 'Hide Details'. Another screenshot shows a table titled 'LinuxFormat Readers' with columns for 'City' and 'Readers'.

City	LinuxFormat Readers
Rome	2800
Milan	1580
Naples	1440
Venice	930
Florence	720
Total:	7470

Other screenshots show the 'Editing Main Page' interface with a rich text editor and various formatting options.

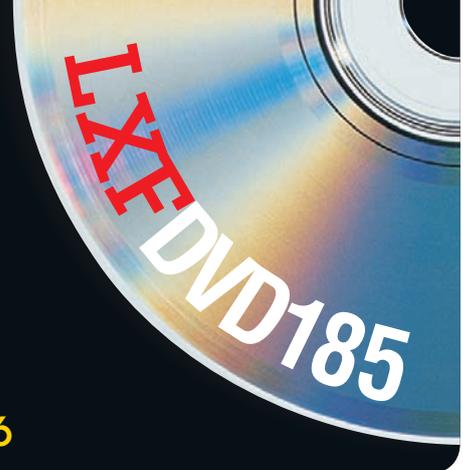
Люди говорят



« Minecraft делает программирование интереснее для детей »

Мартин О'Хэнлон о применении Raspberry Pi учителями с. 40

На вашем бесплатном DVD



Ubuntu 14.04 LTS

» С Unity, KDE, Xfce & LXDE

MX-14

» Antix и Mepis воссоединились

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

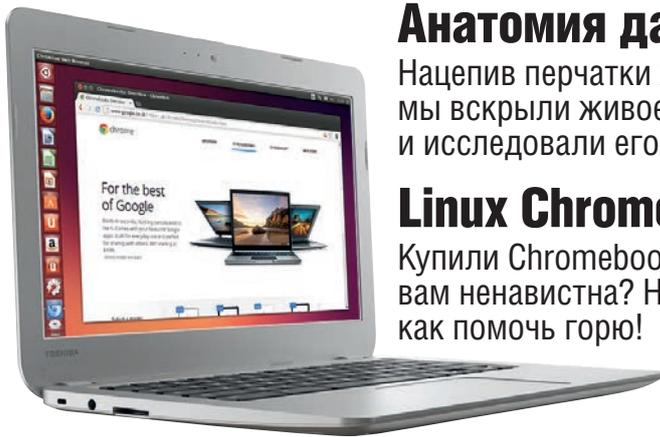
Ищите в этом номере

Анатомия дата-центра 46

Нацепив перчатки хирурга, мы вскрыли живое сердце Интернета и исследовали его биение.

Linux Chromebook 50

Купили Chromebook, но ChromeOS вам ненавистна? Не беда: мы знаем, как помочь горю!



Академия кодига

Python 82, 84

Этот популярный язык программирования осваивается с ходу. Изучим его основы с нуля и пройдемся путями к папкам.

Minecraft Pi 86, 88

Minecraft служит не только для забавы: при посредстве этой

игры можно выучиться программировать на Python. Смеха ради, построим и взорвем дом.

MySQL 92

Заделайтесь волшебником, лихо применяя вместо магического жезла электронные таблицы и MySQL.



Постоянные рубрики

Новости 6

Вышли новые версии ядра Linux, Linux Mint, Red Hat и DragonFlyBSD, ветеран-протокол HTTP омолодили, в Linux шагают целыми штатами, а Россия будет у Samsung'a первой.

Новости Android 22

Samsung готовит планшет в три сложения, студия Kamelot затевает турниры, а китайские смартфоны размножаются со скоростью необычайной.

Сравнение 24

Эти полезные системы управления контентом помогут вам сделать свой сайт на базе wiki: MediaWiki, Dokuwiki, Foswiki, PmWiki, TiddlyWiki.

Интервью LXF 40, 44

LXF расспросил сотрудников британской сферы образования, как правильнее обучать программированию в школах, а Владимир Рубанов

высказался на тему обеспечения цифрового суверенитета страны.

Что за штука... UEFI 54

В погоне за безопасностью ПК великие умы разработали новый способ загрузки компьютера. Не обошлось, понятное дело, и без интриг...

Рубрика сисадмина 56

Доктор интересуется новой системой загрузки в ядро правил фильтрации пакетов, наводит справки в словаре, что такое API и «протокол», и потешается над склонностью протоколов рядиться простачками.

Ответы 96

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик — про извлечение VOB-файлов, безответные клавиатуры, неконтролируемый возврат каретки в скриптах, настройку твердотельных жестких дисков, настройку MIME-

типов в Firefox, ошибки с клавиатурой и мышью в Ubuntu 13.10.

HotPicks 100

Отвечайте горяченького! Лучшие в мире новинки свободного ПО: Assassin Of The Underdark, DispcalGUI, Git, GluCad, GogglesMM, Midnight Commander, Nuttx, Pioneer, Python, Qtractor, Shellpic.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

А ведь он очень удался... Но не страдать, его еще можно заказать, и мир снова заиграет всеми красками!

Через месяц 112

Курочим Ubuntu, готовимся к экзаменам LPI, озираем лучшие дистрибутивы-легковесы и многое другое.

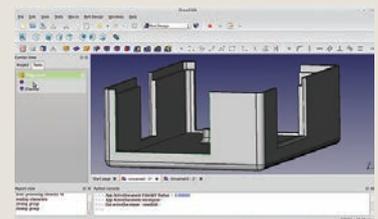
Учебники

Терминал Архивируем с Tar 62

В Linux имеется немало разнообразных архиваторов, но этот — самый популярный.

На помощь! Map-страницы 64

Спасательный круг далеко искать не надо — все предусмотрено здесь же, в системе.



» FreeCAD: создаем и печатаем корпус Pi.

Рационализация Скрипты оболочек 66

Пусть за вас вкальвает компьютер, он железный. Надо только кое-что ему подсказать.

3D-печать FreeCAD 70

Срочно требуется дизайнерский корпус — уберечь Raspberry Pi? Напечатаем сами!

OpenStack DevStack 74

Создадим себе виртуальное облако из виртуальных водяных капель. Ну типа того.

Открытые Evernote NixNote и Etherpad 78

Построим свою альтернативу Evernote для работы над совместными записями.



ГЛАВНОЕ Обновление ядра » Протоколы и спецификации » Тесты драйверов
» 64 бита » Внедрения » Дистрибутивы » Tizen » Red Hat

ЯДЕРНАЯ ВЕСТЬ

С новым ядром вас

Вышел релиз ядра Linux 3.15.

Линус Торвалдс представил очередную релиз ядра Linux. Релиз интересен тем, что в него вошло больше всего изменений за всю историю разработки. Но ждать от него заметных функциональных новшеств не стоит, поскольку этот результат был достигнут за счёт принятия небольших доработок.

Немного сухих цифр:

- » в новую версию вошло более 12000 правок и добавок от 1400 разработчиков, изменения сделаны в 11 428 файлах;
- » размер патча — 57 МБ, добавлено 932 468 строк кода, удалено 571 846 строк;
- » 44% изменений связаны с драйверами устройств, 18% касаются специфичного для аппаратных архитектур кода;
- » 12% изменений затронули сетевой стек;
- » 4% изменений внесено в модули, управляющие файловыми системами;
- » 4% изменений вошло во внутренние подсистемы ядра.

Ощутимо уменьшилось время входа и выхода для спящего режима: в ядре задействованы асинхронные потоки для фаз этого процесса. Хотя на практике пользователи открывают и закрывают крышку ноутбука явно не чаще раза в час. Вообще, конечные пользователи заметят не очень много новшеств. Статус релиза-рекордсмена важен для них сугубо теоретически.

Смешанный режим EFI позволяет загружать 64-битное ядро на 32-битной UEFI. Область применения довольно экзотическая: новое оборудование поставляется с 64-битной UEFI. Но на ряде ноутбуков ещё возникает проблема совместимости.

Адресное улучшение — поддержка клавиатуры Lenovo ThinkPad Carbon, улучшения в драйверах для ноутбуков Sony и Toshiba и т.д. Это очень актуально: ведь речь идёт о лучших ноутбуках в своём классе, которые не могут полноценно эксплуатироваться пользователями Linux.

Это было проблемой для владельцев новых машин, где манипулятор TrackPoint не оснащён аппаратными кнопками — вместо них используются выделенные области кликапада. Приходилось отказываться от эргономичного устройства или подолгу заниматься настройкой.

Реализована начальная поддержка GPU NVIDIA Maxwell и сжатия видео VCE 2.0 на новых видеокартах Radeon. Объявлено о реформах в открытых драйверах. Правда, по качеству им пока далеко до проприетарных. Так что и это изменение скорее из разряда условно полезных.

Геймеров порадует поддержка манипулятора Sony Dualshock 4 из поставки консоли Sony Playstation 4.

Уровень решаемых проблем показывает достаточную зрелость системы Linux. Практически все важные проблемы уже решены, и разработчики по большей части занимаются второстепенными.

ПРОТОКОЛЫ И СПЕЦИФИКАЦИИ

Сетевой ветеран помолодел

Через 15 лет обновился протокол HTTP/1.1.

Комитет IETF (Internet Engineering Task Force) выпустил обновленные спецификации HTTP/1.1. На её подготовку у специалистов ушло семь лет, в течение которых было выпущено 26 черновых вариантов. В результате в итоговые документы включено более 2500 изменений и устранено более 500 недоработок.

Обновления разработала группа HTTP-Bis, которая занималась также и стандартом HTTP/2.0. Для удобства использования спецификации разделили на группы отдельных RFC, чтобы была возможность

ссылаться на технологии, оставшиеся без изменений. Также сообщается об исключении из текста некоторых формулировок, допускавших различные толкования.

Наиболее заметные изменения:

- » снято ограничение на два одновременных подключения к серверу;
- » прекращена поддержка HTTP/0.9;
- » стандартизован заголовок Forwarded для сохранения IP-адреса исходного запроса;
- » удалено требование по использованию ISO-8859-1 как кодировки по умолчанию.

Making It Better

So, in 2007, we started a new Working Group in the IETF called HTTPbis to improve the spec. Our charter was just to clarify, not to change it: we wanted to write the HTTP/1.1 specification down in a better way, not add new features or introduce a new version of the protocol.

Roy Fielding and Julian Reschke led the efforts as editors, splitting up the specifications into seven parts, later to be condensed into six. While doing so added to the page count (since a lot of boilerplate got repeated), it made the drafts easier to read, as it was more logically organised. The resulting specifications are:

- RFC7230 - HTTP/1.1: Message Syntax and Routing - low-level message parsing and connection management
- RFC7231 - HTTP/1.1: Semantics and Content - methods, status codes and headers
- RFC7232 - HTTP/1.1: Conditional Requests - e.g., If-Modified-Since
- RFC7233 - HTTP/1.1: Range Requests - getting partial content
- RFC7234 - HTTP/1.1: Caching - browser and intermediary caches
- RFC7235 - HTTP/1.1: Authentication - a framework for HTTP authentication

Along the way, we issued 26 draft versions, with over 2600 changesets, to address over 550 issues. Discussions of those issues brought in people from Open Source HTTP implementations like Apache, Curl, Squid, Traffic Server, HAProxy and Varnish, as well as people from companies that implement and use HTTP, from Microsoft and Apple to Yahoo! and Google.

» В новой редакции протокола HTTP/1.1 появилось множество изменений.

ТЕСТИРОВАНИЕ

Какой открытый драйвер лучше

Сайт Phoronix опубликовал результаты тестирования 50 GPU.

Известному сайту Phoronix исполнилось десять лет. Это знаменательное событие администрация портала решила отметить масштабным тестированием различных GPU, работающих под управлением открытых драйверов. Первоначальный план предполагал, что в тестировании будет задействовано 65 различных устройств, но на практике получилось меньше — «всего» 50 видеоадаптеров, основанных на Intel HD Graphics, AMD Radeon, AMD FirePro и Nvidia GeForce.

Результаты тестов оказались предсказуемы: открытые драйверы пока не могут на равных конкурировать с проприетарными. И перспективы запуска «тяжёлых» игр на полностью свободных системах остаются достаточно туманными.

Пользователям видеокарт Nvidia следует выбирать открытый драйвер, если им нужны разве что спецэффекты рабочего стола. Разработчикам постоянно не хватает ресурсов, адекватная помощь со стороны вендора отсутствует, более-менее адекватный контроль качества — тоже.

Поэтому значительная часть тестов показывала неудовлетворительный для

взыскательного пользователя результат по причине отсутствия т.н. реклокинга. В общем, для «тяжёлых» задач открытый драйвер Nouveau использовать не рекомендуется.

«Intel подтвердила звание самой удобной видеокарты для Linux.»

Адаптеры AMD тоже показали себя не самым лучшим образом: недостаточно отработана работа в режиме OpenGL 4.x, отсутствует поддержка оверклокинга (разгона) и т.п. А AMD Radeon R9 290 (Hawaii) на момент тестирования вообще работает без 3D-ускорения.

Причина такого положения вещей заключается в том, что разработчики не считают нужным тратить чрезмерно много сил на поддержку новых высокопроизводительных видеокарт: всё равно пользователь выберет проприетарный драйвер. А большинство геймеров вообще пока не рассматривают Linux в качестве серьёзной игровой платформы.

Лучше всего показали себя адаптеры семейства HD6000 — именно их можно рекомендовать пользователям, принципиально предпочитающим открытые драйверы.

И, разумеется, Intel подтвердила звание самой удобной видеокарты для Linux — с ней у пользователя будет меньше всего проблем. Правда, адаптеры этого производителя для «тяжёлых» приложений не подходят — они всё-таки слабоваты.



Сайт Phoronix протестировал 50 различных видеоадаптеров.

64 БИТА

Новый релиз DragonFlyBSD

Разработчики DragonFlyBSD отказались от 32-битной архитектуры.

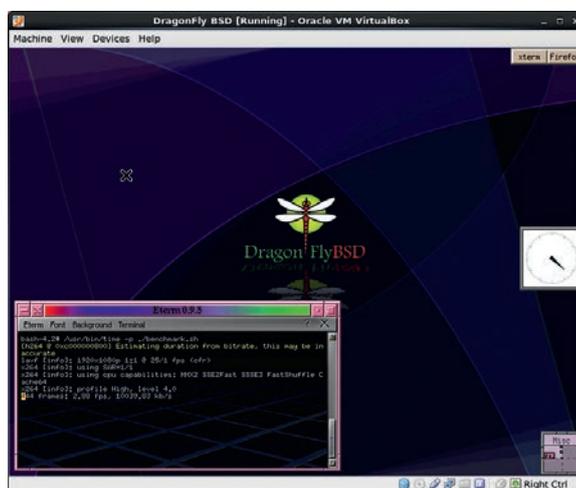
На официальном сайте проекта опубликован очередной релиз DragonFlyBSD под порядковым номером 3.8. Главная его особенность заключается в том, что он стал последним выпуском данной ОС, в котором поддерживается 32-битная архитектура.

Подобное решение разработчиков, вероятнее всего, вызвано тем, что в условиях ограниченных ресурсов они не смогут полноценно поддерживать все теоретически востребованные варианты и сосредотачиваются на наиболее популярном. В настоящее время компьютеры с 32-битной архитектурой — это уже экзотика, сохранившаяся в достаточно бедных учреждениях. У большинства корпоративных и домашних пользователей таких компьютеров уже не осталось.

Поэтому нет никакого смысла делать то, что требуется только единицам или энтузиастам, желающих как-то задействовать купленную десять лет назад машину. При том, что совершенно не очевидно, что кто-то из них выберет именно DragonFlyBSD.

В новой версии дистрибутива пользователя ждут следующие основные новшества:

- » динамические двоичные файлы в каталогах `/bin` и `/sbin`, чтобы обеспечить поддержку современных методов идентификации;
- » драйвер DRM Driver, взятый из ядра Linux 3.8, который пока поддерживает только графический адаптер Ivy Bridge HD 4000;
- » обновление драйвера Radeon DRM, код которого взят из Linux 3.9.



Новая версия DragonFlyBSD — последняя с поддержкой 32-битной архитектуры.

ВНЕДРЕНИЯ

В Linux — всем штатом

Индийский штат Керала планирует перейти на Linux.

Правительство индийского штата Керала, находящегося на юге страны, планирует осуществить срочный перевод всех государственных учреждений на свободное ПО. В качестве основной причины этому называется прекращение поддержки операционной системы Windows XP.

Это решение Microsoft лишило сторонников решений компании главного козыря — экономии на обучении персонала при миграции. Переход с одной версии Windows на другую в обязательном порядке влечёт за собой эти расходы, что сводит на нет главное препятствие внедрению Linux.

Поскольку выход обновлений для Windows XP завершился ещё в начале апреля, за месяц до этого правительство Тамилнада — другого южного штата Индии — начало рассматривать миграцию на свободное ПО как одно из возможных решений возникших проблем. В процессе обсуждения были рассмотрены все

аргументы как за, так и против принятия этого решения.

В пользу перехода на Linux говорило следующее:

- » экономия на покупке лицензий;
- » экономия на приобретении нового оборудования в связи с высокими аппаратными требованиями новых версий Windows;
- » значительно лучшая поддержка индийских языков;
- » поддержка биосенсорного оборудования и цифровых подписей;
- » отсутствие затрат на покупку антивирусных программ.

В качестве замены Windows был выбран дистрибутив BOSS Linux (Bharat Operating System Solutions), основанный на Debian. Он наиболее полно отвечал всем требованиям государственных служб.

В правительстве штата Керала приняли решение взять пример с соседей. В соответствующем указе всем государственным структурам предписано осуществить миграцию до наступления необратимых



» В госорганизации штата Керала будет внедряться система BOSS Linux.

последствий. В настоящее время уже существует руководство по внедрению. Правда, оно носит довольно общий характер и оставляет департаментам достаточную свободу действий.

ДИСТРИБУТИВ

Мятная свежесть

Вышел Linux Mint 17.

На официальном сайте проекта опубликована очередная версия одного из самых популярных настольных дистрибутивов — Linux Mint.

Главная особенность текущего выпуска заключается в том, что он открывает новую схему формирования релизов. Теперь в качестве основы они будут использовать исключительно дистрибутивы Ubuntu с длительным сроком технической поддержки. По всей видимости, эксперименты компании Canonical в промежуточных версиях действительно несовместимы с понятием дружелюбного десктопа.

Таким образом, Linux Mint 17 будет поддерживаться до 2019 года. Правда, после выхода Ubuntu 16.04 LTS (и, соответственно, Linux Mint 18) разработчик планирует выпускать только обновления с устранением уязвимостей и серьёзных проблем.

В менеджере обновлений появился раздел для установки новых доступных ядер Linux. Пользователю будут показаны

все доступные ядра, а также справочная информация по ним.

Ещё одно заметное новшество — изменённый интерфейс настройки языка системы. Это сделано для унификации данного модуля во всех рабочих столах системы — Cinnamon, Mate, Xfce и LMDE. Каких-то заметных дополнительных функций в нём нет, но внешний вид программы стал приятнее.

Linux Mint традиционно выходит в двух вариантах — с десктоп-окружениями Mate и Cinnamon. В этих интерфейсах также сойдётся ряд изменений.

Например, в Cinnamon 2.2 из «Системных настроек» исчез урезанный «обычный» режим. Теперь обычный и единственный режим — это как раз тот, что раньше назывался расширенным. Это, безусловно, правильно: теперь все инструменты настроек отображаются в одном окне.

В общем, новая версия данного дистрибутива в очередной раз подтвердила,

что Linux Mint продолжает оставаться одной из самых дружелюбных к пользователю систем. И пока нет оснований полагать, что ожидается пересмотр данной концепции.



» Рабочий стол Cinnamon — один из оригинальных интерфейсов системы Linux Mint.

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

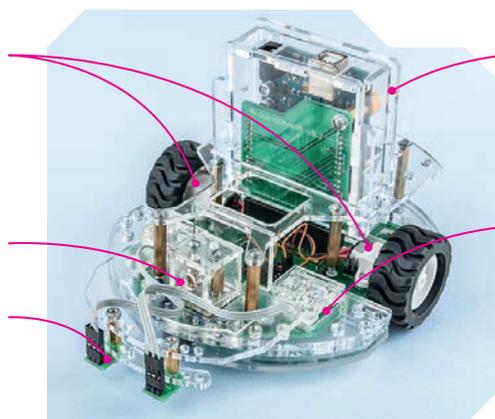
обеспечивает возможность:

- 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)

ДИСТРИБУТИВЫ

Флагман вновь отплыл

Вышел дистрибутив Red Hat 7.0.

Компания Red Hat выпустила очередную версию своего дистрибутива Red Hat Enterprise Linux, рассчитанного на применение в корпоративной среде. Этот релиз был представлен как система, предназначенная для создания гибридного корпоративного облака, гибкая среда для разворачивания и запуска приложений в легковесных контейнерах и обеспечения возможности работы с большими массивами данных без её существенного усложнения.

Компания заявляет, что новый релиз Red Hat даёт пользователю следующие основные возможности:

- » разрабатывать приложения с применением контейнеров Linux (LXC), гарантируя

их быстрое развертывание на физических и виртуальных устройствах;

- » использовать XFS в качестве основной файловой системы;

- » обеспечить межсистемное взаимодействие при использовании доменов Microsoft Windows и Red Hat Enterprise Linux.

В состав системы входят следующие компоненты:

- » *systemd* для управления процессами и сервисами при инициализации;

- » встроенные профили производительности, для оптимизации работы;

- » единый комплекс управления;

- » изолированные контейнеры для обеспечения безопасной работы и тестирования приложений.

TIZEN

Смартфон на букву Z

Россия станет первой страной, где будет продаваться Samsung Z.

Компания Samsung официально представила первый коммерческий аппарат на платформе Tizen — смартфон Samsung Z. Пока эта свободная ОС на базе Linux применяется только в «умных часах» и фотоаппаратах, хотя Samsung уже планирует выпуск телевизора под её управлением. ОС Tizen — по сути, логическое продолжение MeeGo, LiMo. В её разработке участвуют такие крупные компании, как Intel и Samsung, входящие в Technical Steering Group.

На прилавки Samsung Z выйдет в августе текущего года. Пока производитель планирует продавать их только в России: по крайней мере, о других странах ничего не сообщается. Случай уникальный — обычно российский потребитель вынужден приобретать новинки за рубежом.

Технические характеристики смартфона понравятся массовому потребителю. Он никак не выглядит «экспериментальным» или рассчитанным на одних энтузиастов: 4-ядерный процессор с частотой 2,3 ГГц — вполне на уровне современных смартфонов достаточно высокого класса. Как и остальная «начинка» — 2 ГБ ОЗУ и 16 ГБ флэш-памяти. Есть гнездо для microSD. Смартфон имеет 4,8-дюймовый

экран с разрешением 1280×720 — для игр маловато, но для неигрового видео и фото вполне комфортно. Снимки можно делать им же — аппарат оснащён основной камерой на 8 Мпикс. С видеоконференциями проблем тоже не будет — фронтальной камеры на 2,1 Мпикс им хватит.

Ёмкость аккумулятора — скромные 2600 мА·ч. Разработчик сообщал о возможности работы в сверхэкономном режиме, но о его пользе можно будет говорить только после независимых тестов.

Функции коммуникаций стандартны для аппаратов такого класса — Wi-Fi, GPS, Glonass, Bluetooth, USB 2.0, NFC; не хуже, чем у конкурентов. Заодно Samsung Z оборудован датчиком сердцебиения, барометром, сенсором отпечатков пальцев и инфракрасным передатчиком — этим могут похвастаться далеко не все.

Короче, аппарат очень мил. Но говорить о появлении на рынке реального конкурента системы Android пока рано: вспомним судьбу некогда перспективной платформы Maemo. Футурологический прогноз должен быть максимально осторожным. **LXF**

» Смартфон Samsung Z ориентирован на российский рынок.



Новости короткой строкой

» Член управляющего совета Fedora Ярослав Резник [Jaroslav Reznik] официально предложил заменить менеджер пакетов по умолчанию *Yum* на *DNF*. Источник: lists.fedoraproject.org

» Вышла новая версия Manjaro Linux, дистрибутива на базе Arch Linux для начинающих пользователей. Источник: manjaro.org

» Сервис Steam продает пользователям Linux одну из лучших пошаговых стратегий, *Civilization V*, за 25%. Источник: steamcommunity.com

» Опубликована вторая бета-версия пользовательской оболочки Plasma 5, на платформе KDE Frameworks 5 и библиотеке *Qt 5* с OpenGL/OpenGL ES. Источник: kde.org/announcements/

» Пользователям ALT Linux, Fedora, Kubuntu, openSUSE и ROSA доступен выпуск KDE SC 4.13.2., где исправлено 40 ошибок и обновлены переводы. Источник: www.kde.org/announcements

» Программе *Eugene Goostman* от Евгения Демченко (Россия) и Владимира Веселова (США) первой из всех удалось пройти тест Тьюринга. Источник: www.independent.co.uk

» Вышел очередной релиз программы *Docker* для управления изолированными Linux-контейнерами, что позволяет автоматизировать запуск приложений в изолированном окружении. Источник: [https://github.com/docker/](https://github.com/docker)

» Разработчики Lubuntu отложили миграцию на десктоп *LXQt*, создаваемый объединёнными усилиями участников проектов *LXDE*, *Razor-qt* и *Hawaii*. Источник: <https://wiki.ubuntu.com/>

» Компания Google сообщила о выпуске обновления платформы Android под номером 4.4.3 и подготовке прошивок для устройств Nexus. Источник: www.xda-developers.com

» Проект Mozilla объявил о запуске сайта webcompat.com, для сбора сообщений о проблемах в браузерах и сайтах, для их совместного решения. Источник: <https://hacks.mozilla.org/2014/>

» Новый лидер проекта Fedora — участница технического комитета Fedora Engineering Steering Committee Робин Бержерон [Robyn Bergeron] из Red Hat, отвечающая за облачный стек. Источник: robyn.io/2014/

softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



**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

LINUX FORMAT

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Mint и его Cinnamon

За последние годы мы привыкли к тому, что в нашем дивном мире Open Source не происходит ничего; тихо и незаметно выходят новые версии офисов и десктопов, *tar*'ов и *grep*'ов. И потому, когда мы сталкиваемся действительно с событием — оказываемся как бы в растерянности. Не сразу понимаем, что же на самом деле произошло.

Именно таким событием стал выход Mint'a 17-й версии в сборке с Cinnamon'ом (нет, тот, который с Mate, тоже хорош, но скучен). И вроде бы ничего не обещало сенсации (да её и не было). Ну, очередная версия дистрибутива — пусть и самого популярного по рейтингу Distrowatch'a; но мы-то с вами знаем, чего стоит этот рейтинг. Ну и очередная сборка десктопа (кстати, вышедшего за месяц до того). А вот вместе они создали удивительное ощущение гармонии, другого слова не подберу.

На моей памяти в нашем мире произошло два ключевых события. Первым был выход первой версии Mandrake (как ни странно, за номером 5.1). Вторым — появление Ubuntu 5.10, первой ОС Linux, пригодной к реальному применению. Выход Mint'a 17-го — в том же ряду. С Cinnamon'ом, который станет лучшим десктопом всех времён и народов.

И уж простите, в этой колонке довольно много эмоций — и вовсе нет технических подробностей. Но в технических терминах такое не описывается: это надо видеть своими глазами.
alv@posix.ru

Сегодня мы рассматриваем:

- Ubuntu 14.04 LTS** 14
Мы вникли в свежий дистрибутив от Canonical — Trusty Tahr, с долгосрочной поддержкой. Обновиться ли?
- LPS 1.5.1** 16
Дистрибутив на LiveCD, превыше все-го ставящий безопасность в Сети — не от кого-нибудь, а от ВВС США.
- Sonic Pi** 17
Эта забавная музыкальная программа помогает изучить язык программирования Ruby.
- diskAshur Pro** 18
Превосходный внешний жесткий диск с USB 3.0 принимает безопасность данных всерьез и комбинирует физическую и программную защиту.
- AOC A2472PW4T** 19
Монитор с сенсорным экраном вдобавок еще и содержит компьютер на Android.
- Razer BlackWidow** 20
Мы наложили руки на механическую клавиатуру, спроектированную для игр и точного набора.
- FRITZ!Box 7490** 20
Новый представитель линейки беспроводных роутеров продемонстрировал хорошую скорость.
- Обзоры игр** 21
На сей раз мы поиграли сразу в две: головоломку *Circuits* (нет, это не про электросхемы) и пошаговую стратегию *The Last Federation*.



➤ **AOC A2472PW4T:**
ПК «все-в-одном» на ОС Android.



➤ Предполагается, что LTS-релизы должны быть стабильны, но не отошел ли Trusty Tahr от этого правила?



➤ Поочередно делайте ходы, спасая Солнечную систему в этой пространственной космической стратегии.

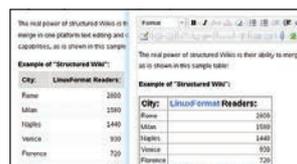
Сравнение: Платформы для википедий

с. 24

Dokuwiki



Foswiki



MediaWiki



PmWiki



TiddlyWiki



Собираетесь запустить свой личный сайт на основе wiki? Мы проверили пять систем управления контентом, которые помогут вам успешно — и с пользой — похоронить Википедию.

Ubuntu 14.04 LTS

Уставший от релизов с долгосрочной поддержкой **Маянк Шарма** без особых надежд приступает к релизу 14.04 — ждет ли его сюрприз?

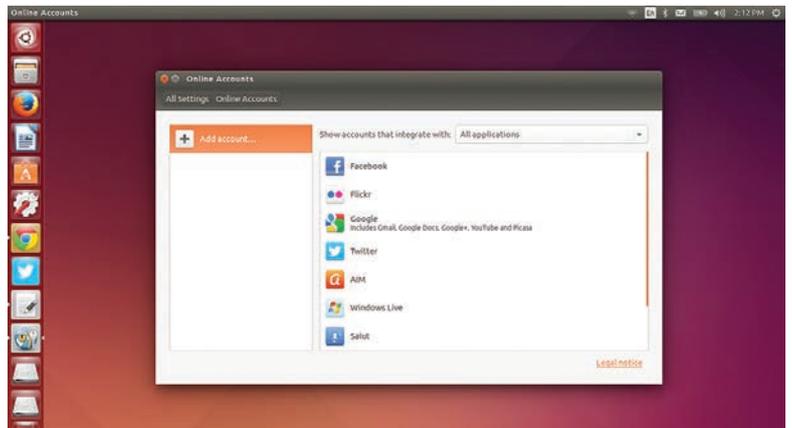
Вкратце

» Один из наиболее современных и популярных дистрибутивов рабочего стола. См. также: Mageia, Mint, Fedora.

Как по часам, Canonical выпустил пятый релиз LTS точно в срок, два года спустя после предыдущего. Количественно это 20-й релиз проекта, вышедший к 10-й годовщине существования. Релизы LTS поддерживаются 5 лет, с основным упором разработчиков на сглаживание острых углов и обеспечение поставки стабильного и свободного от ошибок дистрибутива. Поэтому большинство изменений в LTS покажутся вам по своей природе довольно консервативными.

На первый взгляд Trusty Tahr [Надежный Тар] (кодвое имя) укладывается в такую схему: выглядит вполне себе аналогично предыдущему релизу и не содержит незавершенных компонентов, типа графического стека *Mir*, намеренного продвинуть дистрибутив в его многостаночное будущее. Однако вы вскоре осознаете в поведении рабочего стола малозаметные отличия от прошлых версий. Визуально он очень похож на Ubuntu 13.10, с такими же обоями и без добавочных иконок на программе запуска приложений — *Launcher*. Но недавно внезапно отменили службу облачного хранения *Ubuntu One*, что есть позор. А в *Dash* недостает новых линз.

На самом деле, для понимания нового революционного добавления к дистрибутиву вам придется взглянуть в сопроводительные заметки к релизу [Release Notes] — впервые после появления *Unity* пользователи смогут решать, где поместить меню приложений. По умолчанию они отображаются вверху



» В Ubuntu 14.04 введено много улучшений удобства, таких как новая быстрая блокировка экрана, подобная окну входа в систему.

панели Общее меню [Global Menu]. Но теперь можно перенести их ближе к приложению, перетаскив переключатель *Menu Bar* под *System Settings > Appearance > Behaviour* [Системные настройки > Внешность > Поведение]. Эта функция релизована разумно, и меню показывается в той же адресной строке, где сидят кнопки управления окном. Причем добавление меню не пересекается с перемещением окон, предусмотренным по умолчанию, путем захвата адресных строк.

Находясь в системных настройках окна, имейте в виду, что теперь можно изменять масштаб иконок в *Launcher* вплоть до 16 пикселей. Это несущественное, но полезное добавление, особенно при работе с дистрибутивом не на сенсорном

дисплее с ограниченным пространством экрана. Другая мелкая малозаметная функция — чекбокс в секции *Sound*, позволяющий повисить звук выше 100%.

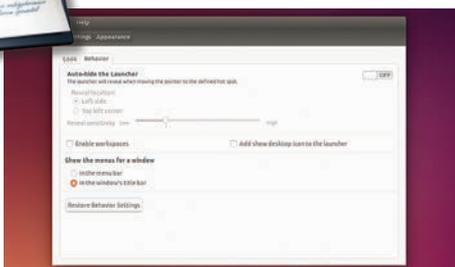
Уплата долгов

Некоторые изменения как в видимых, так и в невидимых компонентах дистрибутива произошли из стремления Марка Шаттлворта [Mark Shuttleworth] употребить этот релиз для уменьшения того, что он называет «технический долг», и для подготовки основы для будущих разработок

Явный пример этого — применение *Ubuntu Browser*, написанного на *QML*. Этот браузер используется в сборках для телефонов и планшетов как часть идеала унификации после *Unity*. На рабочем столе этот браузер используется для отображения *web-приложений*, которые теперь открываются внутри одиночных окон браузера. Тем не менее, браузер не очень хорошо интегрирован с рабочим столом. Так, при поиске документов на диске *Google* с помощью *Dash* те открываются в браузере *Firefox* вместо *Ubuntu*. Более того, в браузере *Ubuntu* не хватает многих, уже ставших привычными функций — например, вкладок. Также новое приложение аномально ведет себя при навигации по открытым страницам и отказывается переключаться между несколькими экземплярами. Нас немало удивило, что в Canonical решили включить в релиз LTS это очень сырое приложение.

Зато дистрибутив как никогда быстр. *Ubuntu* переключились от использования

Свойства навскидку



Локальные меню

Наконец вы получили возможность выдергивать меню из глобального меню и интегрировать их в сами окна.



Больше контроля

Отныне *Ubuntu* дает пользователям больше контроля благодаря новому приложению настройки *Ubuntu*, потому что *Gnome Control Centre*.

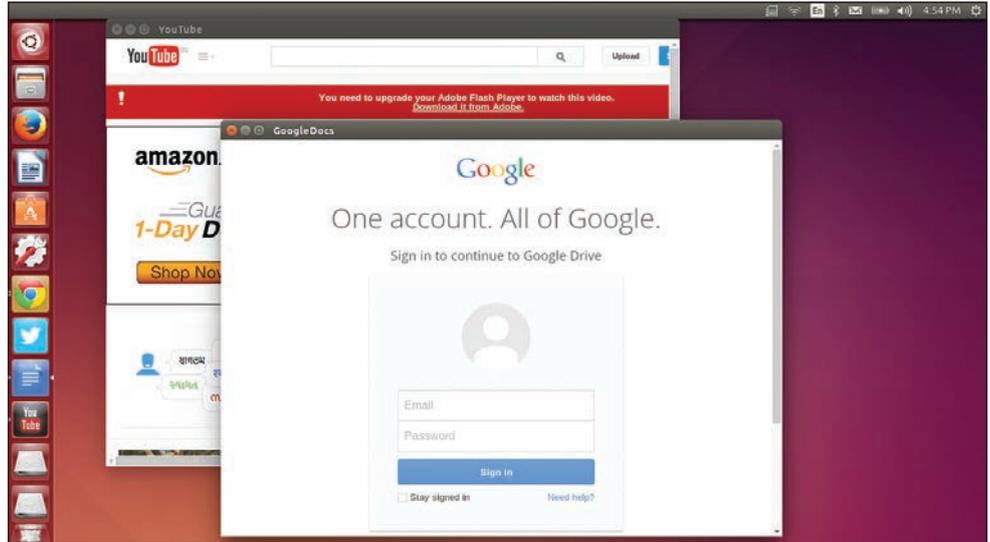
плагина для оформления окон Compiz Decor на оформление окон темами GTK3 CSS. Благодаря этому окна теперь сглажены и закруглены по углам. Черные рамки окон шириной в 1 пиксель теперь убраны.

Прямо перед вашим носом — еще одно изменение, но вы можете не заметить его, пока не приметесь менять размеры окон. В отличие от предыдущих релизов, где был желтый прямоугольник, показывающий размер окна при изменении, в Ubuntu 14.04. действие происходит в реальном времени. Когда вы тянете окно за край и растягиваете до нужного размера, окно тут же расширяется.

Если вы работаете с несколькими окнами, гляньте на небольшое улучшение удобства при активации Spread view. Для непосвященных — это вид, запускаемый по щелчку на интересующем вас значке приложения в Launcher. Если открыто множество окон, Spread view отображает их все. Начиная с Ubuntu 14.04, можно фильтровать открытые окна по их заголовку в Spread view. Просто переключаете вид и начинаете печатать для вызова панели поиска, похожей на Dash и фильтрующей окна по заголовку.

Другой давно желанной функцией, неохотно предоставленной-таки пользователям, хоть и не напрямую, стала возможность уменьшить запущенное приложение щелчком на его значке в Launcher. Для включения этой функции придется скачать приложение *CompizConfig Settings Manager* через *Software Center* и перейти в панель Launcher под секцией Unity, где перечислен перечень функций.

Напомню вам: это релиз LTS, и он пестрит мелкими улучшениями. Улучшена поддержка для дисплеев высокого разрешения, а для SSD-дисков от Intel и Samsung по умолчанию включено TRIM; это должно



» О, смотрите, web-браузер стал другой!

уменьшить падение производительности при длительном использовании таких дисков.

По части графики, дистрибутив поставляется с последней версией драйвера nvidia-prime, который может похвастаться улучшенной поддержкой для технологии оптимизации Nvidia Optimus. Также в релизе есть обещание LTS — пять лет важнейших обновлений и настроек системы

самые поздние версии основных приложений. Это *LibreOffice 4.2.3*, *Firefox 28*, *Thunderbird 24.4*, *Shotwell 0.18*, *Rhythmbox 3.0.2* и *Empathy 3.8.6*.

Дела обстоят так, что Ubuntu 14.04 будет, наверное, последней версией, выпущенной только в настольном варианте. Это также может быть одной из причин, почему Trusty Tahr, возможно, самый шустрый из всех LTS-релизов. LTS не является идеальной платформой по перебору мостов между разными платформами, и поэтому Trusty Tahr являет собой причудливую смесь. С одной стороны, отсутствуют функции, важные для многоплатформенной стратегии Ubuntu — например, сервер отображения Mir; а с другой стороны, включено нечто недоделанное, вроде браузера Ubuntu.

Тем не менее, Ubuntu 14.04 — прекрасный, стабильный дистрибутив без непредвиденных сюрпризов. Теперь рабочий стол Unity дает пользователям больший контроль над его внешним видом и обеспечивает новый взгляд от пользователей, покинувших дистрибутив, когда тот отказался от Gnome. **LXF**

«Прекрасный дистрибутив без непредвиденных сюрпризов.»

безопасности и постоянный механизм стека аппаратного обеспечения [Hardware Enablement Stack], помогающий извлечь максимум из вашего железа.

В отношении ПО, как и в случае с каждым релизом Ubuntu, Trusty включает



» Освоение проходит с той же элегантно простотой, как и было всегда...

LINUX FORMAT **Вердикт**

Ubuntu 14.04 LTS

Разработчик: Canonical
 Сайт: www.ubuntu.com
 Лицензия: GPL и другие

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

» Способен удержать на крючке существующих пользователей, и должен показаться привлекательным совершеннейшим новичкам.

Рейтинг 7/10

LPS 1.5.1

Запустите легковесную ОС Linux с CD, максимально обезопасив свой web-серфинг. Так делает и в том отчитывается отважный **Дункан Эванс**.

Вкратце

» Загрузка ОС Linux с непереписываемого носителя обеспечивает безопасный web-серфинг и безопасную пересылку. См. также **Tails**.

Спецификация

- » Загружается с беспроводного носителя
- » Внутренние диски не монтируются
- » Набор установленных утилит
- » Внутренняя файловая система размещается в ОЗУ
- » Есть почта IMAP и *Firefox* с Java и Flash
- » Шифруется DNS-трафик

» Делает безопасные банковские переводы и извлекает преимущество из программ для безопасности файлов, например, *Encryption Wizard*.

Онлайн-безопасность под угрозой... и, как недавно показала уязвимость Heartbleed, с осторожностью нельзя переборщить. Это суждение лежит в основе системы безопасности *Lightweight Portable Security*, сделанной и выданной в бесплатное пользование научно-исследовательской лабораторией ВВС США для запуска с непереписываемых носителей, т.е. вредоносное ПО или web-вирусы не смогут занести себя на жесткий диск и заразить машину. Между сеансами система перезагружается, и любая подхватившая зараза будет стерта из памяти, обеспечив безопасность следующего сеанса работы в браузере. Это особенно полезно для банковских операций онлайн или при работе с особо важными сайтами.

Скачайте версию *Public Deluxe* с сайта лаборатории <http://spi.dod.mil>; появится обычный ISO-образ, прожигаемый на оптический диск. Затем загрузитесь с CD/DVD на ПК или Mac. Загрузка с CD занимает несколько минут, но вы сразу попадете на рабочий стол. Можно создать и версию для загрузки со съемного устройства памяти, но все равно потребуется загрузка с CD/DVD, а затем форматирование устройства. Загрузка будет впятеро быстрее, но менее безопасна, ибо на носитель можно записать вредоносную программу. Избегайте записи на него персональных файлов: потом они могут быть стерты.

После загрузки смонтируются все диски, подключенные по USB, но не внутренний диск. Создастся файловая система для локального хранилища файлов, но оно целиком помещено в ОЗУ, и после перезагрузки все, что сохранялось, будет потеряно. Можно сохранить файлы на съемном носителе (менее безопасный вариант) или вложить их в электронные



» Система безопасности *Lightweight Portable* предназначена для применения военными, но ее преимущества очевидны и обычным пользователям.

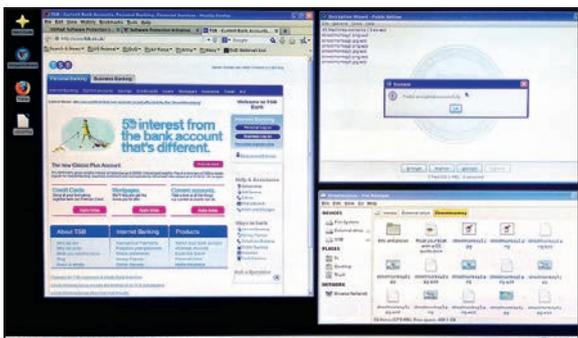
письма и отправить самому себе. Интернет-соединение осуществляется автоматически через Ethernet, а Wi-Fi надо будет настроить утилитой управления сетью.

Большое преимущество возможности выключить сессию или перезагрузиться — зачистка всей сетевой активности: предыстории браузера, личных данных, информации для входа, паролей и т.п. То есть любые клавиатурные шпионы или вредоносные программы из системы вышвырнуты. *Firefox* в комплекте поставки поддерживает Java, Flash и смарт-карты для доступа к защищенным сайтам правительства. Также установлены просмотрщик PDF, *mPaint*, *Unzip*, текстовый редактор *Leafpad*, файловый браузер, *Encryption Wizard*, SSH-клиент и ПО удаленного рабочего стола (*Citrix*, *MSoft* или *VMware*). Для IMAP-сервера почты есть приложения *Thunderbird* и *DAVmail*, а если ваш провайдер применяет POP3, обходной путь — обратиться к web-почте.

В версию *Deluxe* вошел *Encryption Wizard* — одно из ориентированных на безопасность приложений. Пользоваться им легко: перетаскиваете файлы, которые надо зашифровать, в окно интерфейса и задайте пароль на доступ к ним. Затем зашифрованные файлы отправляются получателям электронной почтой. Для создания и последующего шифрования документов также установлен *LibreOffice*. В папке **Security** есть еще пара других связанных с безопасностью приложений. Одно — инструмент, подтверждающий,

что просмотр DNS выполняет *OpenDNS*. Другое, *DNSCrypt*, по умолчанию не активно. Оно активируется при каждом сеансе и использует сервис *OpenDNS* для DNS-просмотров и затем шифрует DNS-трафик. Это может предотвратить ряд атак и усилить конфиденциальность.

LPS не обновляется и не исправляется сама; ВВС США ежеквартально выпускают обновления, требующие новой зачатки и записи диска. LPS — быстрая и эффективная система, но каждый раз учинять полную перезагрузку — решение, слишком затратное по времени, не для повседневной работы. А вот если вы имеете дело со сверхсекретными данными или сайтами — система предлагает новаторский подход к онлайн-безопасности. **LXF**



LINUX FORMAT Вердикт

LPS

Разработчик: AFRL
Сайт: <http://spi.dod.mil>
Лицензия: GPL

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

» Обеспечивает защиту от вредоносного ПО и более безопасный web-серфинг, но для повседневной работы чересчур ограничена.

Рейтинг **8/10**

Sonic Pi

Похоже, эта чудесная среда разработки внушила **Лесу Паундеру** пугающие коллег мечты о блестящей музыкальной карьере.

Вкратце

» Среда программирования, ориентированная на создание музыки с использованием языка Ruby. См. также: Scratch и DIN is Noise.

Есть масса способов обучения программированию, но не все даются в удовольствие. Традиционно, источником знаний служат толстые тома — разве что вам исключительно повезёт с преподавателем. Raspberry Pi Foundation (www.raspberrypi.org) недавно запустили новый сайт, а также комплекс учебных ресурсов, призванных сделать изучение вычислительной техники и программирования веселее. В их числе особо выделяется Sonic Pi, который сочетает в себе удовольствие от создания музыки с логикой программирования.

Sonic Pi — это новый способ знакомства с программированием, эффективно использующий язык Ruby. Ruby очень прост и понятен, а также славится богатой документацией.

Проще простого

Интерфейс Sonic Pi тоже прост и состоит из трех панелей. Слева расположены восемь рабочих пространств, предназначенных для написания кода. Справа вверху находится панель вывода, где построчно отображаются результаты выполнения этого кода. Наконец, справа внизу есть панель ошибок, которая услужливо сообщает пользователю об их наличии. При всей простоте и ясности интерфейса Sonic Pi — из-за чего кое-кто считает его пустоватым — отсутствие отвлекающих факторов помогает пользователям сосредоточиться, позволяя им создавать музыку

» Ruby — простой язык, дающий хорошее представление об азах программирования.

и незаметно для себя учиться немного программировать. После того, как пользователь написал свою музыку/код, достаточно нажать на кнопку play, которая, как и кнопка stop, расположена в левом верхнем углу меню.

Для создания своего шедевра вы можете использовать любую из рабочих областей, но прежде чем уйти и приступить к делу, придётся сделать ещё один шаг. В данном случае, это — сыграть ноту:

```
Play 60
```

Введя эту команду, а затем нажав на кнопку play, вы это сделаете. Число 60 задаёт тон для ноты До. Затем вы можете повозиться с кодом и проиграть первую строчку из *Twinkle, Twinkle, Little Star*.

```
play_pattern [60,60,67,67,69,69,67]
```

Как мы упомянули, Sonic Pi работает на языке программирования, именуемом Ruby; его синтаксис похож на Python. Стандартные концепции программирования, такие как циклы, итерации и параллелизм, легко выстраиваются и предоставляют широкие возможности для написания музыки. Всего за 15 минут нам удалось с помощью Sonic Pi создать простой цикл, состоящий из повтора нот 60–80:

```
for i in (60..80)
  with_synth "pretty_bell"
  play i
  sleep 1
end
```

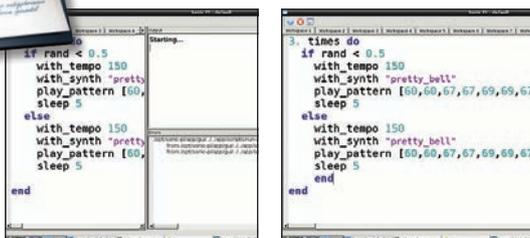
Sonic Pi — не просто приложение: целый ряд ресурсов представлен на партнёрском сайте Компьютерной лаборатории

в Кембридже (<http://bit.ly/SonicPi>). Это прекрасные обучающие ресурсы, призванные дополнить основной учебный план по информатике 3-й ступени, который планируется вводить в британских школах с сентября 2014 года.

Sonic Pi — идеальное введение в программирование, способное перекрыть разрыв между приятным для глаз Scratch и более богатым функциями Python. Музыка — превосходный способ привлечения интереса пользователя, а решение использовать Ruby весьма разумно, с учётом понятного синтаксиса. Было бы здорово, если бы к данному проекту прилагались ещё и учебные материалы, ведь это поможет популяризировать идею, что программирование может быть вполне доступным и творческим. **LXF**



Свойства навскидку



Чистое окружение

Чистое и ненавязчивое окружение позволяет на 100% сосредоточиться на программировании, ни на что не отвлекаясь.

Полезные библиотеки

В Sonic Pi есть также библиотека звуков и других эффектов, которые можно применять при создании композиций.

LINUX FORMAT Вердикт

Sonic Pi

Разработчик: Sam Aaron
Сайт: <http://bit.ly/SonicPi>
Лицензия: MIT

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

» Простой и уникальный способ обучения программированию. Использование музыки в качестве стимула — хорошая зацепка.

Рейтинг **9/10**

diskAshur Pro

Дэвид Хейвард утверждает, что это самый потрясающий USB-накопитель, который вам когда-либо случалось использовать — стоит проверить!

Вкратце

» Владей Джеймс Бонд супер-надёжным USB-накопителем, им был бы diskAshur Pro. Уровень защиты данных в нём — на миллион.

Спецификация

- » Функции: безопасный доступ к данным с помощью клавиатуры, буфер 128 МБ, питание по шине, USB 3.0
- » Габариты: 19×84,5×120 мм, вес 190 г
- » Гарантия: 2 года
- » Сертификаты: NIST FIPS 140-2 уровня 2, FCC&CE, водонепроницаемый

» USB-кабель встроен, так что устройство по-настоящему переносное и кроссплатформенное.

Во сколько вы оцениваете защиту своих цифровых документов? Иным людям безопасность бесценна, в особенности если им приходится перевозить или хранить особо секретные данные. Но даже домашним пользователям стоит присмотреть себе какое-нибудь цифровое хранилище.

У iStorage имеется решение, которое отвечает всем возможным требованиям, и даже больше. Прославившись проектированием супер-безопасных и сверхпрочных портативных жестких дисков, они дарят спокойствие тем, кто регулярно имеет дело с транспортировкой цифровой информации.

Данная версия — diskAshur Pro на 1 ТБ — весьма впечатляющее решение ваших проблем с безопасностью. Во-первых, вы получаете портативный жесткий диск размерами 19×84,5×120 мм и весом всего 190 г. Встроенный кабель USB 3.0 имеет длину 120 мм, а усиленный и обрезиненный корпус обладает свойствами гидроизоляции и невиданной для других портативных жестких дисков прочностью.

Главной особенностью и, пожалуй, главным коммерческим аргументом является наличие крупной цифровой клавиатуры на передней панели устройства. С её помощью можно настроить администратора и пять уникальных PIN-кодов пользователей до 16 символов. Это, в сочетании с 256-битным AES-XTS шифрованием на военном уровне, означает, что вся хранящая информация будет скрыта от всех любопытных глаз, кроме владельца, знающего PIN-код.

Жесткая безопасность

В дополнение к сказанному, есть еще несколько очень полезных функций, как будто прямо из отдела Q. Так, можно установить уникальный PIN самоуничтожения, ввод которого мгновенно сотрёт все контакты и данные, а затем создаст новый ключ шифрования. Он защищен от попыток взлома, не зависит от ОС и работает

с любой из них, включая тонкие клиенты. Доступна даже SSD-версия, и не нужно никакого ПО. diskAshurPro устоит даже под ударом тока или перегрузкой 1000 G, когда не включен, или 300 G — в работающем состоянии.

Наличие USB 3.0 существенно выделяет эту версию diskAshur на фоне предыдущих, как и буфер на 128 МБ и среднее время поиска в 12 миллисекунд. Скорость передачи данных очень высока, согласно *Bonnie++* под Linux Mint 16: более 28 МБ/с на стандартном USB 2.0 и впечатляющие 115 МБ/с через USB 3.0. На нем даже можно смотреть фильмы, хотя это все-таки перебор. Введя PIN-код, можно быстро передать данные, а затем спокойно выйти, зная, что доступом к информации обладаете только вы.

В комплекте вы получите само устройство, подбитый нейлоновый чехол для переноски, краткое руководство пользователя и USB-удлинитель. Инструкции достаточно легко освоить, и вы моментально привыкнете жить по принципу «подключить — ввести PIN — получить доступ — заблокировать — отключиться».

Беспокоит разве что срок службы клавиатуры. Она, конечно, достаточно прочная и хорошо спроектирована, но что если начнут стираться цифры, или из-за ее сбоя не удастся ввести код? Два года гарантии, предоставляемые iStorage после регистрации вашего diskAshur, в какой-то степени обнадёживают, но, как нам сказали, случись такое после — и данные будут потеряны.

В общем, этот DiskAshur Pro — очень интересная штука, к тому же с разными вариантами объема памяти. Если



» Компактный, прочный, очень надёжный USB-накопитель, который будет служить годами.

безопасность данных для вас превыше всего — хоть при перевозке информации, хоть для создания надёжной резервной копии, или вы по долгу службы связаны с транспортировкой секретных или важных электронных документов — то продукция diskAshur незаменима для вашей организации. Так или иначе, diskAshur, безусловно, стоит попробовать. **LXF**

LINUX FORMAT Вердикт

iStorage diskAshur Pro

Разработчик: iStorage
Сайт: <http://goo.gl/1eECpg>
Цена: от £209 (500 ГБ)

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

» Безопасность ваших данных бесценна, и diskAshur Pro надёжно укроет их от посторонних глаз.

Рейтинг **8/10**



AOC A2472PW4T

Умнейший Android «все в одном» уже доступен? Проверят Дэвид Хейвард.

Вкратце

» Беглый взгляд в будущее AIO и настольного монитора, или просто другая ниша сенсорных продуктов?

Спецификация

- » База: Android 4.2
- » Доступные пакеты: Google Play + обычные базовые приложения и виджеты
- » Функции: 68-см сенсорный дисплей IPS, разрешение 1080, 4-ядерный процессор 1,6 ГГц Nvidia T33 CPU, ОЗУ 2 ГБ, память 8 ГБ, 2×USB, HDMI, VGA, Ethernet, Wi-Fi, Bluetooth, web-камера 720 пикселей

Если будущее технологии экранов лежит в устройствах «сенсорной ориентации», то AOC уверенно берет его в руки. Вот AOC A2472PW4T, 68-см умный сенсорный монитор «все-в-одном», вместе с ПК на Android 4.2, с разнообразием функций, которые раздражат интерес и энтузиастов, и деловых пользователей.

В современной офисной среде, где работники ждут самую лучшую производительность, мобильные коммуникации и впечатляющую зрелищность, этот умный AIO оправдывает надежды. ЖК-панель имеет родное разрешение 1920×1080 и дает яркий, живой ряд цветов без большой размытости (хотя резкость похуже, чем на «менее-умном» мониторе).

Четырехъядерный процессор 1,6 ГГц Nvidia T33 CPU, в комплекте со впечатляющим 2-ГБ ОЗУ DD3 и флэш-памятью eMMC на 8 ГБ, позволяет этому устройству весьма ускориться, мгновенно сделав его способнее аналогов — типа ViewSonic VSD220 или TD2340 — с точки зрения сенсорных экранов. Добавления — два USB-порта, микро-USB, VGA, HDMI (с поддержкой MHL), Bluetooth, Ethernet, 802.11. b/g/n, передняя web-камера на 702 пикс, читалка карт SDHC и пара встроенных динамиков 2W — в сумме дают приятный набор.

Если нужно сделать некую работу, просто включите питание A2472PW4T в режиме Android и подключите Ethernet, клавиатуру и мышь. По умолчанию никаких приложений для работы не установлено, но не займет много времени просмотреть хранилище Google Play в поисках подручных инструментов вроде *Kingsoft Office*.

Используя умный дисплей, можно полистать Интернет, составить календарь и даже поредактировать, но вы сильно ограничены качеством приложений, доступных для Android. A2472PW4T неплох как рабочее устройство — данный обзор печатался в приложении для обработки текста на подсоединенной клавиатуре — но недостаток общего контроля над мобильными приложениями, в противоположность установленным на Linux, может привести к тому, что устройство будет восприниматься как некая диковина, а не серьезный конкурент стандартному ПК «все-в-одном». Работой оно не ограничивается — можно насладиться обширной коллекцией игр в хранилище Google Play и просмотреть практически любые формы HD-медиа, благодаря впечатляющим характеристикам A2472PW4T. Как и у большинства устройств AIO, динамики крошечные, но добавка приличных настольных динамиков или наушников создает колоссальную разницу.

Вы тронуты

Перегибаться через стол к экрану немного неудобно. Если притянуть его поближе, придется таращиться на огромный экран в нескольких дюймах от собственного носа. Честно говоря, работа с сенсорным экраном доставляет удовольствие только вначале и довольно скоро начинает раздражать. Использование клавиатуры и мыши оказывается даром свыше, и по сути спасает AIO от участи быть просто планшетом-переростком.

К дизайну A2472PW4T есть вопросы. Для начала, отвлекает черная рамка шириной в дюйм; затем следует еще дюйм на самой панели, который отнимает часть экрана и визуально его уменьшает. Он тяжелый и неповоротливый; ножка-опора тугая, и установка ее в удобную позицию довольно неудобна из-за веса продукта. Правда, конструкция хороша; пластиковая



» Он довольно эlegantный, несмотря на некую громоздкость.

окантовка края нигде не скрипит, а отрегулировав ножку-опору, можно быть уверенным, что устройство не рухнет.

Реагирующий на несколько пальцев сенсорный экран работает хорошо и опознает жесты, «мазки» и щипки без намека на проблемы. В качестве экрана для презентаций AIO максимально близок к идеалу, а играть крупным планом на большем экране приятно.

В целом, несмотря на негативные моменты, это хороший сенсорный экран на базе Android. Цена его высоковата для технологии планшета и с учетом спецификаций экрана, но у устройства хороший потенциал. На самом деле, это одно из лучших устройств такого плана из доступных прямо сейчас. **LXF**

В качестве монитора

AIO A2472PW4T — хороший монитор. Регулировки высоты нет, настраивается только наклон, но этой встроенной связки достаточно для использования.

С системой Windows 8 сенсорный экран работал очень хорошо, различные дистрибутивы на базе Ubuntu также выказали превосходную производительность. Жаль, что резкость монитора не такая, как хотелось бы. Уровни цвета

и яркости были прекрасны, но в повседневных задачах, которые мы выполняли, в целом не доставало детальности, резкости и четкости линий.

Коммерческой приманкой, очевидно, служит объединение режима Android и сенсорного экрана, но сенсорные экраны остаются узкой нишей, и чтобы широкая публика к ним основательно привыкла, предстоит еще много работы.

LINUX FORMAT Вердикт

AOC A2472PW4T

Разработчик: AFRL
Сайт: <http://goo.gl/kANfKq>
Цена: £ 359

Функциональность	8/10
Производительность	7/10
Удобство использования	6/10
Оправданность цены	7/10

» Хотя есть несколько потенциальных проблем, это интересный взгляд в будущее устройств «все в одном».

Рейтинг **7/10**

Razer BlackWidow

Оправдана ли столь высокая цена за механическую клавиатуру? Анализирует **Фил Иванюк**.



» Эта клавиатура не только удобна, но и привлекательна с виду и даже на слух.

LINUX FORMAT Вердикт

Razer BlackWidow

Производитель: Razer
Сайт: www.razerone.com
Цена: £ 115

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

» Отлично смотрится, удобная в использовании клавиатура, но отсутствуют функции, имеющиеся в подобных товарах за схожую цену.

Рейтинг **8/10**

Спецификация

- » Ширина 475 мм
- » Высота 171 мм
- » Глубина 20 мм
- » Вес 1500 г
- » Звуковой выход/гнездо микрофона
- » USB-порт

Несмотря на кошмарно высокую цену, клавиатура Razer BlackWidow Ultimate — штука относительно простая. Здесь нет режима мигания ЖК-диодов или внешних дисплеев, обычно встречающихся у клавиатур, предназначенных для игр: только прямые линии, равномерная и эффективная зеленая подсветка, пять спецклавиш с макросами и прочная — в принципе, почти бомбоупорная — сборка.

Матовое резиновое покрытие корпуса клавиатуры новым выглядит классно, но долго это не продлится, если вы склонны поедать чипсы за игрой: пятна и кляксы к нему липнут. Клавиатура не оснащена упором для рук, но используя подставку на ее тыльной стороне, довольно легко найти положение, удобное для печатания. Плетеный USB-кабель позволит осуществить транзитную передачу с одного USB; еще есть разъем для аудио и 3,5-мм

вход микрофона, спрятанные с правой стороны клавиатуры для опрятности.

Переключатели под клавишами — собственный дизайн Razer, но если вы приняли вызов от Pepsi и набор переключателей от Cherry MX Blue, надо быть уж очень большим виртуозом печати, чтобы ощутить разницу. Звук, сопровождающий нажатия клавиши, изменяется по мере перехода от касания к максимальному упору. В отличие от некоторых механических клавиатур, способствующих многочасовым опечаткам (или неудачным броскам гранатой в играх), мы считаем BlackWidow очень точной и хорошо управляемой.

Чего действительно не хватает, так это встроенной регулировки громкости, какая есть, например, в клавиатурах Corsair. Вместо этого придется нажать функциональную кнопку, а с ней и F2, и F3; что выглядит достаточно позорным для такого дорогого товара. **LXF**

FRITZ!Box 7490

Новый роутер FRITZ!Box 7490 хвастает беспроводной AC-совместимостью. **Мэтт Хэнсон** берет его в оборот.



» При, быть может, спорной внешности, FRITZ!Box 7490 — могучий маршрутизатор.

LINUX FORMAT Вердикт

FRITZ!Box 7490

Производитель: AVM
Сайт: www.fritzbox.eu
Цена: £ 247

Функциональность	10/10
Производительность	9/10
Удобство использования	8/10
Оправданность цены	7/10

» Данный роутер 802.11ac предлагает отличную скорость и ряд современных функций, хотя и по цене выше среднего.

Рейтинг **8/10**

Спецификация

- » Беспроводные AC и N с 1300 МБ/с (5 ГГц) и 450 МБ/с (2,4 ГГц) одновременно
- » VDSL и ADSL+
- » 4×LAN на 1 ГБ
- » 2×USB 3.0/2.0 для принтеров и устройств хранения
- » Медиа-сервер для музыки, изображений и видео
- » Функциональность NAS (FTP, SMB, UPnP AV)

Немецкая компания AVM создала впечатляющую репутацию своей линейке сетевых устройств Fritz. Эстетика у них на любителя, зато внутри скрываются очень компетентные сетевые технологии. Такова беспроводная технология 802.11ac, обещающая максимальную скорость 1300 МБ/с и расширенную полосу пропускания для вашей беспроводной сети, если ваша машина оснащена встроенным AC-адаптером. Беспроводная скорость выше, чем у проводного Ethernet-соединения. Звучит слишком хорошо, чтобы быть правдой, но в наших тестах 7490 показала впечатляющей пропускную способность 371 МБ/с, когда наш ноутбук, оснащенный AC (Intel Dual Band Wireless-AC7260 с драйверами Linux), был в одном помещении с роутером. А когда мы вышли из комнаты и поднялись на один пролет лестницы, скорость снизилась до все еще очень даже приемлемых 130 МБ/с.

FRITZ!Box 7490 поставляется с рядом современных сетевых функций, которые отчасти оправдывают высокую цену. Они могут одновременно транслировать двойные беспроводные сети AC и N в обоих диапазонах, 5 ГГц и 2,4 ГГц. Порты USB 3.0 позволяют разделить через вашу сеть доступ к принтерам и устройствам хранения, для этого служат 4 гигабитных порта LAN. Встроенное ПО предлагает массу инструментов настройки, а-дажем, родительский контроль, хотя интерфейс у него не самый дружелюбный к пользователям из нами виденных. Все же, он легко доступен через web-браузер с любого подключенного устройства. Также 7490 работает как базовая станция DECT для шести телефонов — удобно для малого бизнеса.

Все эти современные инструменты делают 7490 несколько дороже стандартного маршрутизатора, так что крепко подумайте, нужны ли они вам. **LXF**

Circuits

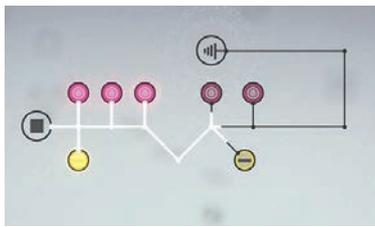
Мэтт Хэнсон берётся за новую музыкальную головоломку.

Вкратце

» Музыка и головоломка смешаны в одном флаконе ради увлекательной и уникальной игры, подобных которой мы и вправду не видели — и не слышали.

» Чтобы решить загадки в *Circuits*, создавайте музыкальные треки.

Большинство головоломок нацелено на тренировку умственной деятельности, заставляя вас чесать затылок в попытках сообразить, как же уложить эти частички и перейти на новый уровень. Однако *Circuits* — не из их числа. По названию — или походя глянув на экранный снимок — легко заключить, что речь идёт о создании электрических цепей. В реальности же тут нет ничего общего с электроникой, и вместо сообразительности в решении головоломок вы должны полагаться на свои уши. При загрузке вам сообщают, что для достижения наилучших



результатов следует использовать наушники — и это отнюдь не шутка. Каждый уровень состоит из короткого музыкального трека, который надо внимательно прослушать и который также визуальнo представлен в виде печатной платы, где инструменты, ноты и аккорды изображены отдельными компонентами. Затем необходимо воспроизвести эту дорожку, разместив иконки вдоль цепи. Начитается всё с простого, но кода дело доходит до сложных треков на нескольких инструментах и слоях, задача усложняется. Хотя объёмная музыка не всем по душе, исполнена она хорошо и, в сочетании с минималистской графикой, производит успокаивающее и расслабляющее воздействие. Первые несколько уровней позволили нам разобраться с концепцией геймплея, не надрывая свой потрёпанный рок-фестивальными слух. Но потом наш интерес довольно быстро увял. В малых дозах — это уникальный и расслабляющий способ

скоротать пару минут, но далеко не самая захватывающая игра из нами виденных. К счастью, разумная цена, скромных £2 [порядка 120 руб.], превращает её во вполне лакомый кусочек. **LXF**

LINUX FORMAT Вердикт

Circuits

Разработчик: Digital Tentacle
Сайт: <http://bit.ly/DigTentacle>
Цена: £ 1,99

Сюжет	6/10
Графика	7/10
Продолжительность	6/10
Оправданность цены	7/10

» Если вы заядлый меломан и любитель головоломок, денежки потратить не жаль.

Рейтинг **6/10**

The Last Federation

Солнечная система в опасности! Вся надежда на Мэтта Хэнсона...

Вкратце

» Пошагово сражайтесь со своим противником не на жизнь, а на смерть — или заключайте союзы на безграничных просторах галактики.

Восстановить мир в Солнечной системе, где миллиарды жителей готовы воевать с ближайшими небесными соседями, задача непростая. На первый взгляд миссии *The Last Federation* исчерпываются стрельбой и маневрированием на своем корабле против врага, но можно также попробовать решить конфликты путём дипломатии — хотя и здесь есть подвох. *The Last*



» Далеко не «Автостопом по Галактике», но улыбку вызовет.

Federation — пошаговая стратегия, а значит, вы по очереди определяете, куда двигать свой корабль и какие вражеские корабли атаковать (если таковые имеются). А затем ваш противник делает то же самое — как в шахматах. Однако, в отличие от шахмат, ходов другого игрока вы не видите, пока не сделаете свой — ходы совершают одновременно, а затем следят за развитием конфликта (или отсутствием такового). Атакуя, можно навести мышью на корабль противника, чтобы посмотреть его защиту и решить, какое оружие нанесёт больше повреждений. Это придает схваткам глубину — хотя можно закончить уровень и пойдя на мировую. Что подразумевает стыковку с вражеским судном и передачу ресурсов, которые могут использовать против вас же; но зато улучшится ваша репутация в стане противников, впоследствии превратив их в союзников. Сюжет у игры захватывающий и очень продуманный, графика и музыка

достойные (хотя и не сногшибательные). В целом игра напоминает старые пошаговые стратегии — это тоже неплохо. А доля юмора помогает не принимать её слишком всерьёз. **LXF**

LINUX FORMAT Вердикт

The Last Federation

Разработчик: Arcen Games
Сайт: www.arcengames.com
Цена: £ 14,99

Сюжет	8/10
Графика	6/10
Продолжительность	7/10
Оправданность цены	7/10

» Забавная и на удивление сложная космическая пошаговая стратегия с хорошим чувством юмора.

Рейтинг **7/10**



SAMSUNG

Планшет со складным экраном

Компания Samsung готовит необычный планшет.

Несмотря на то, что рынок мобильных устройств развивается весьма динамично, каких-то радикально новых устройств на нём пока не особенно заметно. Поставят более мощный процессор, более ёмкую батарею... Или экран с высоким разрешением. Но не более того.

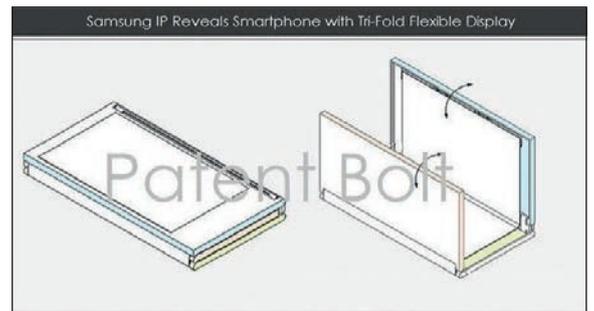
Согласно данным, которые получил сайт Sammobile из инсайдерских источников, в ближайшее время пользователей может ждать действительно интересная новинка. Компания Samsung готовит к выпуску необычный планшет, экран которого будет складываться в трёх местах. Предполагается, что его анонс состоится в следующем году.

Инсайдеры также сообщают, что компания Samsung не только запатентовала этот планшет, но и выпустила несколько тестовых образцов. На закрытом показе,

проходившем в рамках выставки MWC 2014, ограниченному кругу лиц показали два планшета с OLED-дисплеями, размеры которых составляли восемь и девять дюймов.

Интересна в устройстве только его компактность — в сложенном виде оно вряд ли будет заметно больше обычного смартфона, который спокойно умещается в кармане. По полученной сайтом Sammobile информации пока неизвестно, будет ли работоспособен частично разложенный планшет.

Поэтому вряд ли стоит ждать каких-то чудес. Даже если всё это соответствует действительности, то Samsung, вероятнее всего, выпустит всего-навсего «диковину» для гиков, готовых заплатить большие деньги за обладание тем, чего больше ни у кого нет. Практический же смысл новинки особо не просматривается.



» Возможно, скоро пользователи увидят планшеты со складывающимся экраном.

Здравый смысл подсказывает, что по крайней мере первый тираж устройства будет крайне ограничен, и скорее всего большая его часть разойдётся «по своим» — программистам, журналистам и блогерам. Если какая-то партия и дойдёт до прилавков, то цена будет абсолютно неприемлемой для большинства пользователей.

ИГРЫ

Новая игра от Gameloft

Пользователи Android могут поучаствовать в рыцарских турнирах.

Знаменитая в геймерской среде компания Gameloft порадовала пользователей Android новой интересной и, что немаловажно, бесплатной игрой *Rival Knights* [Рыцари-соперники]. Как следует из названия, игроку предстоит пройти долгий путь и одержать немало побед на турнирах. Понятно, за неплохую награду — несметные сокровища короля.

Первое, что бросается в глаза — игра требует вертикальной ориентации планшета, при которой его значительно удобнее держать одной рукой. Таким образом, любители путешествовать лёжа в купе смогут коротать время не только чтением или преферансом.

Графика игры достаточно реалистична, причём сочетается с полным отсутствием «тормозов». Очень неплохо проработана

физика, придающая игре высокую степень реалистичности: в поединках буквально чувствуется вес каждого элемента.

Суть игры сводится к рыцарским поединкам и прокачиванию персонажа. Врагов у рыцаря хоть отбавляй, поэтому скучать не придётся. Однако имеется один не совсем приятный момент.

Для участия в битве необходимы королевские печати, которые приходится тратить по штуке на поединок. Восстанавливаются они относительно долго — минут за пять. Или — покупайте печать за реальные деньги... поэтому бесплатная игра будет только для терпеливых.

Желающие сразиться с другими игроками тоже должны использовать печати — только другие. Принцип там тот же: или жди, или покупай.



» *Rival Knights* — новая игра от студии Gameloft.

Справедливости ради заметим, что на доспехи и оружие хватает заработанного в процессе игры. Видимо, авторы учли, что в те старые добрые времена торговцы-жадюги долго не жили, так что цены в магазине вполне демократичные.

ZTE

Смартфон Nibua Z7

Новые подробности о планшете компании ZTE.

Китайская компания ZTE хорошо усвоила уроки правильного PR и обставила выход своего флагманского смартфона Nibua Z7 массой слухов и «утечек». До недавнего времени считалось, что аппарат будет оснащён экраном Full HD с диагональю 6,44 дюйма. Из-за таких размеров его даже успели окрестить «планшетом», поскольку он явно выходил за границы уже привычной всем «лопаты».

Смартфон пока ещё не вышел. Зато появились новые сведения о нём.

Оказывается, «планшетофона» не будет. А будет обычная пятидюймовая «лопата» с разрешением 1920×1080. Показатель, кстати, весьма приличный: таких аппаратов на рынке немного.

Что касается начинки, то она достаточно стандартна для недешёвых аппаратов — процессор Qualcomm с тактовой частотой 2,5 ГГц и 3 ГБ оперативной памяти. Операционная система — Android 4.4 KitKat с фирменной графической оболочкой Nubia UI 2.0.

Дешёвым смартфон Nibua Z7 назвать сложно: по имеющимся новым данным, он будет стоить примерно \$450. В этом сегменте рынка конкуренция довольно высокая, и новому товару будет непросто обойти другие бренды.

Не исключено, что именно поэтому компания и решила организовать подобную «рекламную кампанию» и как-то заинтересовать потенциального покупателя. Следует заметить, что ей это удалось —

теперь пользователи хотя бы в курсе, что скоро на рынке появится довольно приличный смартфон Nibua Z7, и захотят покрутить его в руках.



➤ Анонс планшета Nubia Z7 ожидается в ближайшем будущем.

HTC

Смартфон HTC One (E8)

HTC выпускает новый вариант флагманского смартфона.

Компания HTC продолжает развивать линейку смартфонов One. На сей раз она представила общественности новую версию этого флагманского аппарата, получившего название HTC One (E8).

Продажи должны начаться в середине лета этого года. Правда, не во всех странах: пока достоверно известно только про Китай и Тайвань. О цене устройства ничего не сообщается, но она не будет превышать стоимость других флагманских смартфонов линейки.

Поскольку революционными технологическими идеями производители

смартфонов публику особо не балуют, то пользователям всё чаще приходится выбирать товар исключительно «по одежке». Разработчики про это знают и пытаются предложить покупателю что-нибудь интересное.

Внимание покупателя HTC One (E8) должен привлечь корпус из поликарбоната — этот прочный и практичный материал всё чаще используется для изготовления корпусов электронных устройств. В предыдущем варианте смартфона для этой цели применялся алюминий — от дешевого пластика компания отказалась уже давно.

О электронной начинке этого аппарата ходило множество слухов. Но всем гаданиям уже поставлена точка — сейчас известны все характеристики HTC One (E8).

Процессор — четырёхъядерный Qualcomm Snapdragon 801 с тактовой частотой 2,5 ГГц. Память — 2 ГБ оперативной и 16 ГБ встроенной. На сегодняшний день это уже заурядный набор для смартфонов премиум-класса, к которым и относится наш герой. Подобные характеристики позволят владельцу применять аппарат для решения практически любых задач, включая мультимедийные. Кстати, наличие стереодинамиков BoomSound само по себе намекает на широкие развлекательные возможности устройства.

Вместо камеры Duo Camera в новой версии смартфона установлен 13-мегапиксельный фотомодуль с BSI-сенсором. Это уровень весьма продвинутой «мыльницы», поэтому невзыскательные фотографии могут немного облегчить свой багаж.

Для хранения файлов может использоваться сменная карта microSD ёмкостью до 128 Гбайт. Ёмкость аккумулятора составляет 2600 мАч. По сравнению с предыдущей моделью вес устройства снизился на 15 граммов. **LXF**



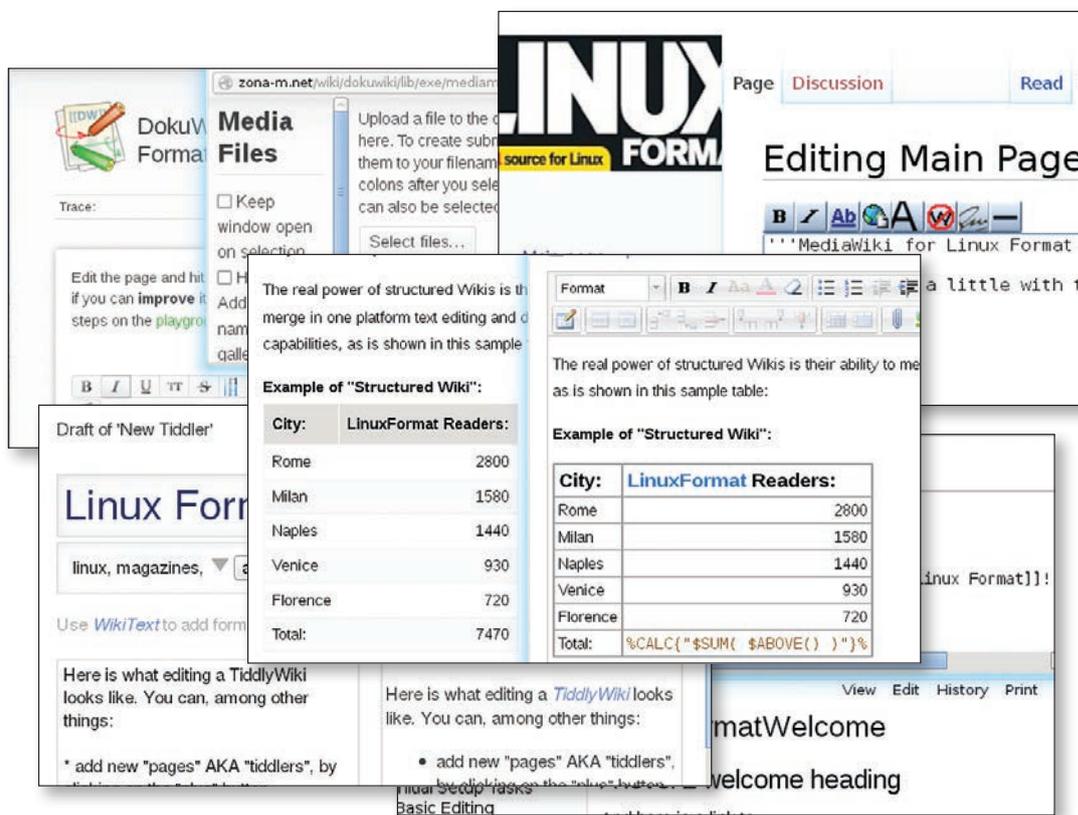
➤ HTC One (E8) — новый флагман известного вендора.

Сравнение

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

Платформы Wiki

Марко Фиоретти нашел пять wiki-платформ СПО для создания и упорядочения массы документов и сотрудничества над ними, чтобы их увидел мир.



Про наш тест...

Wiki свободного ПО очень много. Пять примеров, описанных на этих страницах, были выбраны по двум основным критериям. Мы хотели, чтобы над ними велась активная разработка, что было бы полезно и хорошо и для отдельных личностей, и для больших сообществ, и мы очень хотели показать, насколько разнообразен мир свободных wiki. Именно поэтому в этой статье вы найдете знаменитого тяжеловеса *MediaWiki*; продукт одного пользователя; малоизвестную, но очень мощную структурированную wiki; и две небольших, традиционных wiki, простых в использовании и богатых функциями. Все программы были установлены на виртуальный сервер CentOS 6 с помощью рекомендованных методов, но мы также упоминаем и альтернативы, при их наличии. Затем мы тестировали каждую программу, создавая пользователей и страницы и устанавливая плагины.

Наша подборка

- » Dokuwiki
- » Foswiki
- » MediaWiki
- » PmWiki
- » TiddlyWiki



Факт 1: Хотя может показаться, что мы истинно живем в эру мультимедиа, наши культура и общество по-прежнему основаны на создании и упорядочении огромного количества текстовых документов и предоставлении доступа к ним — как мы ни пытаемся игнорировать сей факт.

Факт 2: Сотрудничество и командная работа зачастую — но далеко не всегда — дело хорошее, и Интернет может сильно способствовать этому.

Столкновение этих двух реалий привело к появлению обширной категории Систем управления контентом [Content

Management Systems, CMS], известных как wiki. Эти пакеты отличаются от CMS типа *WordPress* или *Drupal*, поскольку недвусмысленно предназначены для создания и публикации текстов онлайн — в основном в результате совместной работы — любым заинтересованным лицом, как можно быстрее и с минимумом контроля.

Известнейшая из действующих программ wiki — *Wikipedia*: большинство пользователей, скорее всего, пощещали ее хотя бы раз в жизни, пусть даже по ошибке. Но огромный

успех *Wikipedia* имеет как минимум один контрпродуктивный эффект: многие полагают, что это единственный вариант создания wiki с помощью свободного ПО.

Наше Сравнение опровергнет данное утверждение: сообщество СПО создало массу самых разных wiki, и мы надеемся помочь вам найти именно ту, которая идеально подойдет вам и вашим партнерам.

«Самый известный пример действующей wiki — *Wikipedia*.»

Установка

Трудно ли установить wiki?

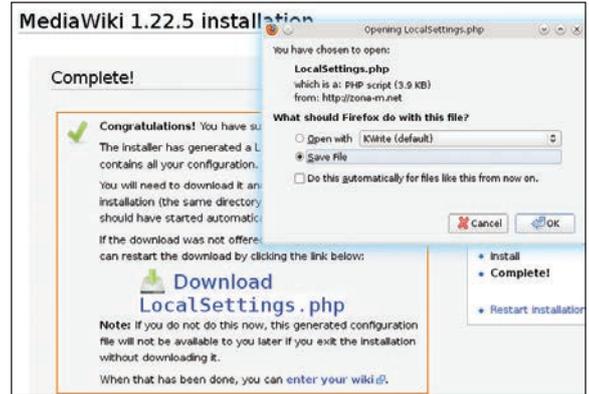
Установка wiki разнится от «очень легкой» до «дьявольски сложной». Например, *TiddlyWiki* вообще не требует настоящей установки: ее единственный файл просто надо сохранить в читаемой директории. В некоторых случаях вам может понадобиться расширение *TiddlyFox* для *Firefox* (или его эквивалента в *Android*), если вы хотите сохранить изменения в локальной файловой системе; но это, собственно, все. Администраторы, желающие централизованно управлять *TiddlyWiki* для разных пользователей, могут установить ее как пакет движка *JavaScript Node.js*.

Установка *DokuWiki* начинается со скачивания, которое позволяет пользователю решить, какой язык и какие плагины должны быть включены. Есть спартанский веб-сервер (только для *Windows*) для работы с *DokuWiki* с диска *USB*. Вам нужно создать подпапки под названием **conf** и **data**, куда может писать веб-сервер, загрузить файл **install.php** в браузер и следовать инструкциям. Закончив это, настройте папку **conf** на режим только для чтения и удалите файл **install.php**, чтобы устранить возможность атак.

Процедура для *PmWiki* очень похожа: создайте подпапку **wiki.d** с возможностью записи, скопируйте шаблон настройки (**docs/sample-config.php**) в **local/config.php**, отредактируйте последний файл, настроив основные параметры, и, наконец, загрузите страницу **pmwiki.php**. *PmWiki* может работать со встроенным веб-сервером.

Установка *MediaWiki* похожа на установку *WordPress*. Перед началом создайте базу данных и учетную запись пользователя для доступа к ней. После проверки среды PHP мастер предложит администратору выбрать движок базы данных из тех, что имеются на сервере. Опциональным — но желательным — шагом будет выбор профилей прав пользователя и настроек электронной почты. Затем надо загрузить файл **LocalSettings.php** для необходимого редактирования и сразу же занести его в главную директорию **wiki**.

Foswiki более сложна в установке и требует доступа **root** к серверу. По возможности, используйте на выбранном компьютере официальные образы виртуальной машины. Двоичные пакеты *Linux* на сайте могут потребовать настройки вручную,



➤ **Финальный шаг мастера установки MediaWiki, использовать который — одно удовольствие.**

чтобы облегчить установку из исходников. В таком случае нужно будет создать файл **.htaccess**, защитить программу настройки *FosWiki* паролем и подготовить индивидуальную настройку. Это должно быть сделано с помощью онлайн-генератора на <http://foswiki.org/Support.Apache-ConfigGenerator>. Назовите получившийся файл **foswiki.conf** и поместите его в поддиректорию, зарезервированную для модулей *Apache* (в *CentOS* — **/etc/httpd/conf.d/**). Только после этого вы сможете загрузить страницу **bin/configure** в свой браузер; она работает более или менее как *MediaWiki*, и создает суперпользователя, который будет заниматься постоянным администрированием.

Вердикт

MediaWiki	★★★★★
TiddlyWiki	★★★★★
DokuWiki	★★★★★
PmWiki	★★★★★
Foswiki	★★★☆☆

» *MediaWiki* обладает наилучшей комбинацией функций и простоты в использовании.

Архитектура

Что происходит за сценой, и почему?

Оценивать архитектуры, создаваемые для разных целей — штука непростая. Говоря в общем, wiki могут хранить контент в простых текстовых файлах, базах данных или системах управления версиями. Wiki на основе файлов менее масштабируемы и/или быстры, чем другие, но зато работают на учетных

записях бесплатного хостинга и проще в управлении и создании резервных копий. *TiddlyWiki* — это одна страничка HTML/JavaScript, предназначенная для одиночных пользователей. Она сохраняет «страницы», именуемые **tiddlers**, и переписывает их контент внутри самой себя. Здесь можно зашифровать все с помощью пароля.

DokuWiki использует PHP, и, вероятно, достигла оптимального компромисса между функциями и простотой: множество пользователей, немедленное начало работы с версиями и темами без всяких баз данных, на любом сервере. То же можно сказать и о *PmWiki*.

MediaWiki, которая также написана на PHP, справляется с большой нагрузкой и разными языками (на ней работает *Wikipedia*). Оптимальными движками будут *MySQL* и *SQLite*, но поддерживаются еще *Oracle*, *PostgreSQL*, *MariaDB* и *Percona*. *MediaWiki* предназначена для того, чтобы максимально упростить обычное, анонимное участие.

Foswiki — программа Perl, способная работать как самостоятельно, так и на веб-сервере. Это структурированная wiki, и в общих чертах это означает, что она создана для поддержки определенного типа высоко структурированного контента — почти как реляционная база данных.



➤ **System Web** и боковая панель по умолчанию в *Foswiki* ошеломительны, что, впрочем, неизбежно, учитывая сложность данного пакета.

Вердикт

DokuWiki	★★★★★
MediaWiki	★★★★★
PmWiki	★★★★★
Foswiki	★★★☆☆
TiddlyWiki	★★★☆☆

» *DokuWiki* и *MediaWiki*, вероятно, осчастливят большее число пользователей.

Интерфейс пользователя

Просто ли с ними работать?

Все wiki в нашем сравнении поддерживают ту или иную форму тем, но, пожалуй, на том индивидуальная настройка интерфейса и заканчивается (если, конечно, вы сами не хакнете их код). С учетом этого факта, основополагающим различием может стать внутренний редактор.

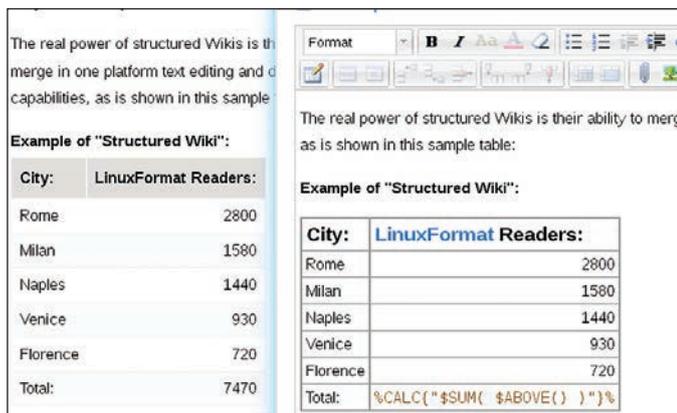
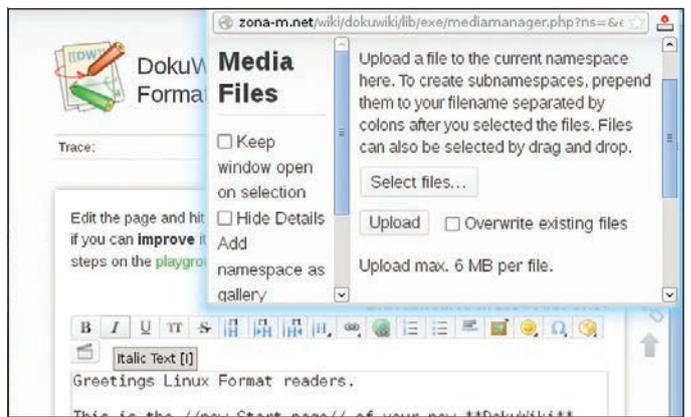
Все wiki используют некую форму текстовой разметки в качестве исходного формата своих страниц. Однако некоторые требуют, чтобы вы сами писали эту простую разметку или использовали обычный HTML.

У остальных имеются кнопки, как в традиционных офисных пакетах WYSIWYG. Некоторые

редакторы WYSIWYG wiki облегчают публикацию для абсолютных новичков, а также позволяют просто копировать и вставлять текст, выбирать случайным образом шрифты и делать многое другое прямо в этих офисных пакетах, получая в итоге непредсказуемый и довольно неприглядный результат. Оно того стоит? Решать вам.

DokuWiki ★★★★★

DokuWiki великолепна в своей простоте. Здесь есть темы, если они вам нужны, а редактор по умолчанию отлично выглядит, и работает не хуже. Если хотите — если вы с ней уже знакомы — можете сразу использовать разметку *DokuWiki* при написании. Например, текст, заключенный в двойной слэш или звездочки, будет отображаться курсивом или жирным шрифтом соответственно. Если вы представления не имеете о том, что такое разметка, тогда просто выделите часть текста, щелкните по кнопке с соответствующим форматированием, и *DokuWiki* добавит простую разметку там, где вам это нужно. Этот процесс делает изучение правил форматирования относительно простым и безболезненным. Но хотим дать совет: прежде чем вставлять изображение, прочитайте простую страничку документации по пространству имен *DokuWiki*, иначе вы можете забыть, как это делается, и в итоге получить битую ссылку на изображение на своей странице.



Foswiki ★★★★★

Foswiki состоит из полунезависимых «паутин [web]», т.е. взаимосвязанных наборов документов. Отдельные страницы называются «тема [topic]»; им можно присваивать тип, как переменным. Шаблоны определяют вид и то, как работает *Foswiki*. По умолчанию есть простой скин для браузеров и скин печати, оптимизированный для бумаги. Помимо редактора, основной частью GUI *Foswiki* является набор инструментов — настраиваемая боковая панель, содержащая все задачи, выполнение которых разрешено данному пользователю. Редактор по умолчанию — WYSIWYG, но также поддерживается язык разметки *Foswiki* и простой HTML. Самое главное — контент каждой страницы до некоторой степени программируется через использование макросов и/или формул, подобных электронным таблицам (как показано на рисунке слева). Все это вместе с макросами, типами страниц и мощными функциями поиска и является тем, что мы называем структурированной wiki.

Опции импорта и экспорта

Если wiki созданы, чтобы делиться, легко ли это сделать?

Wiki существуют, чтобы делиться написанным и полученными результатами. Они отлично подходят для онлайн-публикации готовых документов. Для программ wiki важна простота импорта или экспорта их контента.

В *TiddlyWiki* мало что есть для импорта или экспорта, разве что кнопка Download Me и инструмент командной строки. В *Foswiki* есть плагины для импорта файлов *MediaWiki* или CSV.

Страница *MediaWiki* Special:Export создает «дамп контента» выбранных страниц,

или всей wiki в виде одного XML-файла, опционально включающего шаблоны и предыдущие версии. Расширение *Pdf-Book* сохраняет все статьи в заданной категории в виде одного документа PDF. Другие инструменты, подсказки и хитрости для полуавтоматического массового импорта из других wiki, а также файлов CSV, HTML, OpenDocument или Microsoft Word, имеются на www.mediawiki.org/wiki/Manual:Importing_external_content.

На сайте *DokuWiki* объяснено, как конвертировать документы HTML в принятый

формат разметки. *PmWiki* здесь набирает очки благодаря своим руководствам и/или плагинам для массового импорта из других wiki или блогов *WordPress*. *DokuWiki* и *PmWiki* и без плагинов обгоняют в этом своих более утонченных соперников. Если страницы являются стандартными файлами в обычных папках, web-мастера могут создавать их тысячами с помощью довольно простых скриптов оболочки. На <https://forum.dokuwiki.org/thread/8072> вы найдете обсуждение с примерами подобной стратегии.

Вердикт

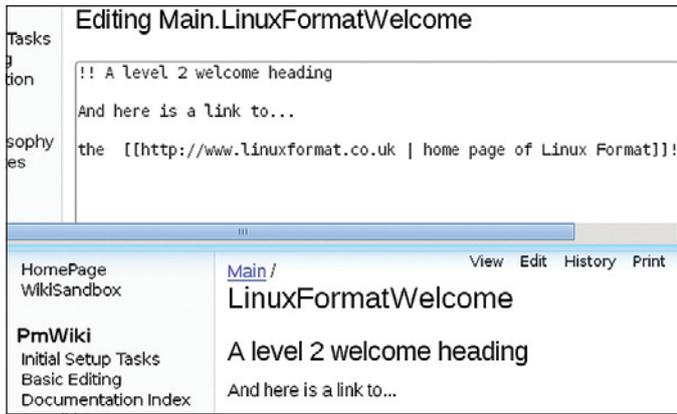
MediaWiki	★★★★★
Foswiki	★★★★★
PmWiki	★★★★★
DokuWiki	★★★★★
TiddlyWiki	★★★★★

» В *MediaWiki* больше плагинов, но основным на файлах wiki они могут вообще не понадобиться.

MediaWiki ★★★★★

Благодаря своим посещениям Wikipedia, вы должны знать, как выглядит «читаемая» часть *MediaWiki*, и о ее поддержке множества языков — однако как насчет ее другой, «редакторской» стороны? Редактор по умолчанию в *MediaWiki* ведет себя так же, как и редактор *DokuWiki*, только кнопок меньше — однако не стоит считать это ограничением.

Помимо тонн руководств и шпаргалок, сильно упрощающих изучение синтаксиса, и без того несложного для понимания, есть еще множество плагинов, которые расширяют способности форматирования *MediaWiki*. Например, один из них, под названием *Poem*, добавляет тэг <poem> специально для облегчения форматирования стихотворений! Другие плагины заботятся о сложных таблицах. Предусмотрено множество опций для форматирования или изменения размера изображений и правильного размещения их в тексте документа.



PmWiki ★★★★★

Лозунг *PmWiki* — «Четкая философия и сопротивление (до некоторой степени) ползучему функционализму». Это отношение отражается во внутреннем интерфейсе пользователя, проще которого и быть не может. В нем есть боковая панель, окно поиска, ссылки на версии с возможностью печати, журнал и обратные ссылки на текущую страницу. Элементы на боковой панели можно добавлять или реорганизовывать, как и списки на любой другой странице wiki. У вас также есть пакеты локализации для множества языков, настраиваемые стили и целый ассортимент графических скинов.

Что касается редактора, здесь нет кнопок или иных отвлекающих моментов — только форма ввода текста. Прямо под формой располагается встроенная шпаргалка со всеми основными правилами и ссылками на «продвинутые» варианты форматирования, например, простые или расширенные таблицы.

TiddlyWiki ★★★★★

Почти официальное определение *TiddlyWiki* — «нелинейный персональный web-блокнот на следующие 25 лет». И в это верится: настолько она просто и легко работает и настраивается, даже для новичков в JavaScript. Несмотря на такую простоту, или, возможно, как раз благодаря ей, интерфейс *TiddlyWiki* работает прекрасно. Он самый быстрый из всей тестируемой здесь пятерки, поскольку практически все вы делаете в браузере, не создавая перегрузок сервера.

Чтобы приступить к работе с *TiddlyWiki*, нужно знать всего пару вещей. Во-первых, «страница» здесь называется tiddler, и чтобы создать новую, нужно щелкнуть по значку с плюсом, чтобы открыть редактор и его окно предпросмотра live. Во-вторых, интерфейс можно настроить, щелкнув по значку с колесиком и вдохновившись галереей на <http://tiddlyspot.com/?page=gallery>.



Документация

Просто ли найти то, что мне нужно узнать?

Для *MediaWiki* документация и поддержка — не проблема. Имеется очень подробный технический справочник и хороший раздел подсказки на www.mediawiki.org, а отдел поддержки на сайте — одно из самых посещаемых мест среди всех wiki.

TiddlyWiki приятно удивляет объемом документации. Вы найдете уйму подсказок на <http://tiddlywikiguides.org>, а на <http://tiddlywiki.org> — еще больше материала, включая дополнительную информацию по разработке. Справочник *DokuWiki* очень

подробный, но программа настолько проста, что он вам, скорее всего, нечасто будет нужен. Периодически начинает казаться, что *DokuWiki* выбрала свое имя, чтобы донести до всех, что все wiki хороши для документации... однако эта еще лучше именно для данной задачи. Загляните в FAQ «Что делает *DokuWiki* настолько идеальной для документации» на сайте, где вы найдете исчерпывающее объяснение.

PmWiki идет с прямой ссылкой на всю свою документацию, включая отличный ситуационный справочник, который

вы сможете прочитать на www.pmwiki.org/wiki/Cookbook/.

У *Foswiki* больше материала, чем вы в принципе готовы прочитать. Немало времени уходит просто на то, чтобы пробраться сквозь индекс. Помимо всего прочего, здесь есть несколько введений и слайд-шоу, руководство пользователя, полный справочник и множество ресурсов для администраторов, включая специальное руководство по инструменту настройки *Foswiki*. При всем ее отличном качестве, это еще одно свидетельство ее сложности.

Вердикт

- Foswiki ★★★★★
- DokuWiki ★★★★★
- MediaWiki ★★★★★
- PmWiki ★★★★★
- TiddlyWiki ★★★★★

» У *Foswiki* больше всего документации, но захотите ли вы ее читать?

«Уход» и контроль доступа

Как держать вашу wiki в достойной форме и под контролем?

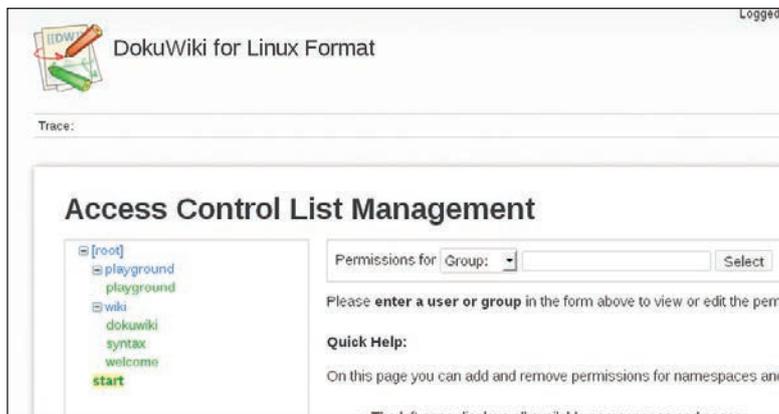
Wiki разработаны так, чтобы быть как можно проще для конечного пользователя. Обеспечить эту простоту несколько сложнее. Администратор wiki обязан постоянно обновлять программу и надзирать за тем, чтобы пользователи могли делать только то, что им нужно, и ничего больше.

Доступ к *TiddlyWiki* почти не контролируется. Ее процедура обновления состоит в обдурировании браузера: загружается старая версия, отменяется ее старый файл и сохраняется новый с тем же именем в вашем менеджере файлов, а затем в браузере нажимается кнопка Save to Disk!

Обновление *PmWiki* или *DokuWiki* — просто замена старых файлов PHP новыми. В *PmWiki* можно установить пароли на разные операции. Эти ограничения можно включить для всей wiki или для определенных пользователей или групп через Access Control Lists (ACLs). Есть плагины для CAPTCHA, черных списков, соединения с движками аутентификации и предоставления доступа внешним ссылкам на URL из белого списка. У *DokuWiki* самый лучший интерфейс для создания пользователей, групп и настройки ACL. Когда она включена, панель web-контроля облегчает установку плагинов.

GUI web-администрирования *MediaWiki* работает через Special Pages [Особые страницы], если не установить сторонние панели управления. Ограничить редактирование просто, зато заблокировать доступ для чтения иначе, чем через профили пользователей в мастере установки, совсем другое дело. *MediaWiki* с самого начала написана для того, чтобы делиться контентом без ограничений. Поэтому, хотя и можно найти плагины, позволяющие заблокировать доступ, вы обнаружите разные полуофициальные предупреждения о том, что они не поддерживаются. Обновления *MediaWiki* несколько сложнее обновления ее основанных на файлах конкурентов. Загрузив новые файлы в их web-папку, надо либо перезапустить программу установки, либо (что предпочтительнее) запустить скрипт PHP из командной строки.

У администратора *Foswiki* проблема та же, что и у документации. Вероятно, есть хоть одна кнопка для любой задачи, которая вам когда-либо понадобится, однако вы замучаетесь ее искать. Даже процедура обновления — хоть и не сложная сама по себе — вызывает головную боль. Вам придется вручную копировать или перемещать множество файлов. И не забудьте о должной защите страницы настройки: пожалуйста, отведите некоторое время исключительно на эту работу.



➤ Настройка Access Control Lists весьма хитроумна, но *DokuWiki* ее облегчает.

Вердикт

- DokuWiki ★★★★★
- MediaWiki ★★★★★
- PmWiki ★★★★★
- Foswiki ★★★★★
- TiddlyWiki ★★★★★

» *DokuWiki* предлагает лучшие функции и безопасность, что возводится в квадрат простотой администрирования.

Плагины

Мне нужно больше функций!

Расслабьтесь: у всех описанных здесь wiki множество плагинов — но ни одна установка не потребует их всех.

Например, форма Node.js *TiddlyWiki* может работать в командной строке, сохраняя небольшие tiddler'ы в формате HTML и выполняя множество других задач. Обычные расширения *TiddlyWiki* идут как tiddler'ы. Чтобы установить их, скопируйте исходный код на их домашнюю страничку в *TiddlyWiki* в качестве новых, ординарных tiddler'ов, и настройте так, как объясняется на <http://tiddlywiki.org>.

Плагины *DokuWiki* надо просто поместить в их собственную папку в `lib/plugins/`. Обычно это возможно сделать с помощью менеджера плагинов из web-интерфейса. Домашняя страница *DokuWiki* также

предлагает комплекты плагинов в качестве готовых «решений для общих приложений *DokuWiki*».

В *PmWiki* имеется множество плагинов и инструментов браузера, включая фильтры LaTeX и интерактивные разделы страниц, но нет единой директории, чтобы перечислить их все. Вместо этого вы должны догадаться, где находится плагин, прочитав ситуационный справочник или страницу функций на www.pmwiki.org/wiki/PmWiki/Features.

На момент написания данной статьи матрица расширения *MediaWiki* на www.mediawiki.org/wiki/Extension_Matrix имела около 2000 пунктов. Установка расширений обычно подразумевает ручное



➤ *DokuWiki* имеет лучший встроенный интерфейс для управления плагинами.

редактирование файла `LocalSettings.php`. *Foswiki* находится почти в той же ситуации, с обширным списком расширений на <http://foswiki.org/Extensions>. Что бы вы ни захотели встроить в *Foswiki*, велика вероятность, что там оно уже имеется.

Вердикт

- Foswiki ★★★★★
- MediaWiki ★★★★★
- DokuWiki ★★★★★
- PmWiki ★★★★★
- TiddlyWiki ★★★★★

» *Foswiki* и *MediaWiki* предлагают множество расширений, которые легко найти.

Платформы Wiki

Вердикт

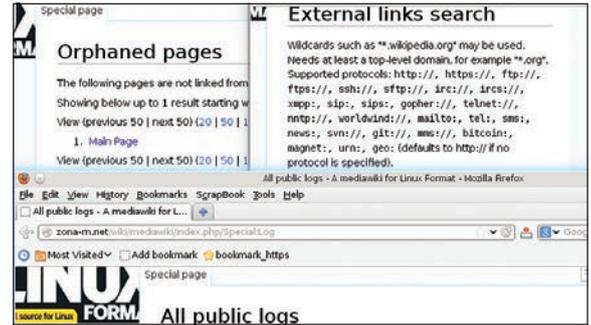
Гонись мы просто за мощью и гибкостью, мы бы немедля объявили победителем *Foswiki*. Это не просто инструмент, позволяющий группам случайных пользователей стать соавторами текста наиболее быстрым и простым из имеющихся способом. *Foswiki* — впечатляющая платформа для создания web-приложений, которые выглядят, как обычные wiki, но которым надо выполнять солидную обработку данных. Однако многим ли людям или организациям реально нужно нечто подобное? Если только вы не убеждены в том, что она вам нужна, изучение управления и использования *Foswiki* явно не стоит затраченных усилий.

Похожие соображения, но под иным углом, исключают *TiddlyWiki*. Это самое простое и быстрое приложение из всех рассмотренных, да и самое гибкое, потому что будет работать на любой ОС — включая Android — даже без интернет-соединения. *TiddlyWiki* заслуживает того, чтобы

ее узнали лучше, но она не предназначена для командной работы над wiki.

Зато *PmWiki* — настоящая wiki, дающая возможность совместного редактирования, простого контроля доступа и истории страниц. Она очень проста в настройке, использовании и поддержке. То же можно сказать о *DokuWiki*, которая имеет больше функций и индивидуального помощника скачивания. А главное, у *DokuWiki* имеется отличный web-интерфейс администрирования, который даже превосходит интерфейс по умолчанию *MediaWiki*. Однако последний имеет больше расширений и лучше подходит для работы с большим количеством пользователей и большой нагрузкой. Функции просмотра журнала и управления версиями *MediaWiki* — самые полные и лучше всего протестированные в группе.

Благодаря ее истории и известности, в *MediaWiki* имеется



уйма функций мониторинга и отчетности, очень важных для тех, кто хочет настроить действительно публичную wiki — со всем, что полагается, включая решение проблем с вандалами, троллями и т.п. Поддержка, и отчеты о деятельности, и логи *MediaWiki* сообщат администратору все, что надо знать для пресечения атак войны редактирования, позволив также находить страницы, давненько не обновлявшиеся. Все вместе, эти факторы и сделали *MediaWiki* победителем, но с очень малым отрывом: ведь *DokuWiki* — классный инструмент!

Здесь показана лишь малая часть способностей мониторинга *MediaWiki*.

«*TiddlyWiki* заслуживает, чтобы ее узнали лучше, но она не для командной работы.»

I

MediaWiki ★★★★★

Сайт: www.mediawiki.org Лицензия: GPL Версия: 1.22.6
» Мощная и расширяемая, с лучшими возможностями мониторинга.

IV

PmWiki ★★★★★

Сайт: www.pmwiki.org Лицензия: GPL Версия: 2.2.63
» Понятная и приятная в использовании, но более ограниченная, чем ее конкуренты.

II

DokuWiki ★★★★★

Сайт: www.dokuwiki.org Лицензия: GPL Версия: 2013-12-08 "Binky"
» Кому не нужна масштабируемость и функции отчетов *MediaWiki*, выбирайте ее!

V

TiddlyWiki ★★★★★

Сайт: <http://tiddlywiki.com> Лицензия: GPL Версия: 5.0.10-beta
» Быстрая и симпатичная. Крута, но не для многопользовательских сценариев.

III

Foswiki ★★★★★

Сайт: <http://foswiki.org> Лицензия: GPL Версия: 1.1.9
» Этой мощной и впечатляющей wiki нужны не менее мощные пользователи.

Обратная связь

Мы забыли упомянуть вашу любимую wiki? Присылайте свои мнения об этом Сравнении на ixf.letters@futurenet.co.uk.

Рассмотрите также...

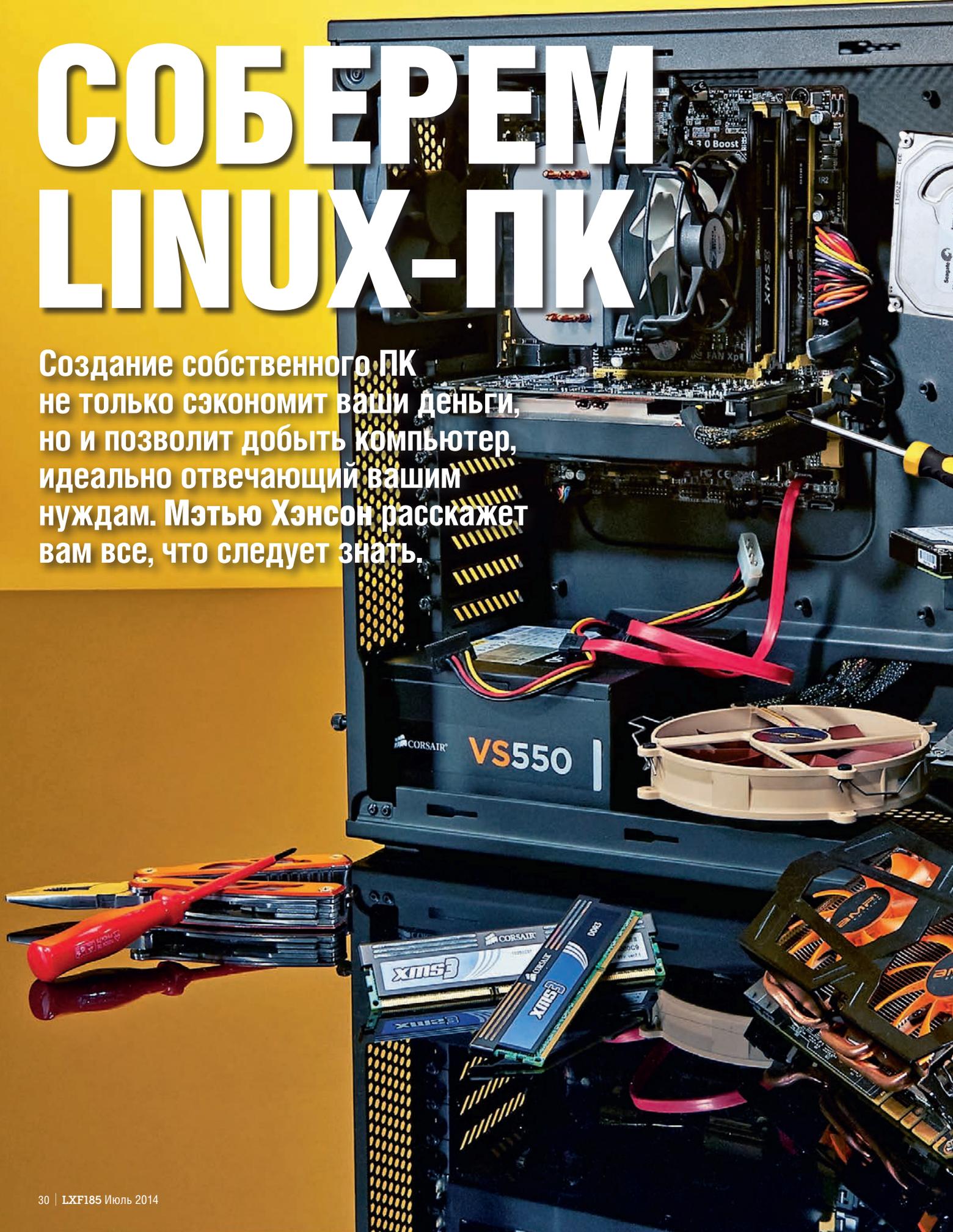
Эти пять очень разных wiki были выбраны сознательно, чтобы показать разнообразие ландшафта мира свободных wiki. Есть и много других программ wiki, и вам незачем ограничиваться описанными здесь. Загляните на www.wikimatrix.org и используйте их мастер, чтобы осознать масштаб вашего выбора. Что до *Foswiki*, «20-минутное руководство [20 Minute Tutorial]» (<http://foswiki.org/>

[System/TwentyMinuteTutorial](http://foswiki.org/System/TwentyMinuteTutorial)) позволит вам узнать, нужна ли подобная структурированная wiki вашей организации. Если вы по-прежнему не уверены, подумайте о том, сколько времени вы можете посвятить управлению своей wiki по сравнению с реальным ее использованием. Если время, умения и прочие ресурсы ограничены, наилучшим выбором может стать wiki на базе файлов.

Она позволяет поиграть с разрешениями файлов и облегчит задачу защиты вашей wiki от атак. Еще один неплохой способ оценить wiki — начать с ваших самых экзотических требований. Вам придется вставлять много математических формул или публиковать много сложных таблиц? Посмотрите, у какой wiki имеется лучшая комбинация редактора и плагинов для упрощения этих задач. LXF

СОБЕРЕМ LINUX-ПК

Создание собственного ПК не только сэкономит ваши деньги, но и позволит добыть компьютер, идеально отвечающий вашим нуждам. Мэтью Хэнсон расскажет вам все, что следует знать.





Приобретение для установки Linux готовой машины может казаться самым удобным способом, но покупая компьютер в магазине, вы идете на целый ряд компромиссов. Главный из них — цена: вы платите некую сумму тому, кто собирает для вас ПК. Это разумно для тех, кто хочет малость облегчить себе жизнь. Но в таком случае вы мало контролируете — если контролируете вообще — использованные компоненты, и можете даже переплатить за то, что в принципе вам не нужно. И это подводит нас ко второму компромиссу: оборудование. Хотя некоторые производители ПК позволяют перед покупкой индивидуально подобрать компоненты вашего компьютера, все же эти опции часто ограничены. А значит, вы никогда не получите полный контроль над производительностью машины, и в итоге у вас окажется машина, идеально подходящая для одних задач, но убогая для других. И последний основной компромисс, на который вы вынуждены идти, отлично знаком нам, пользователям Linux: ПО. До сих пор на удивление трудно купить готовый компьютер без Windows — или Mac OS X, если вам это больше по нраву — и кучи ненужных предустановленных программ. Очистка жесткого диска и установка выбранного вами дистрибутива уже само по себе лишние хлопоты, но это также означает, что вы в итоге заплатили за лицензию на выброшенную ОС.

Собирая собственную машину, вы не идете на эти компромиссы. Вы никому не платите за сборку и не раскошеливаетесь на крутые бренды — чтобы увидеть где-нибудь название «Sony» или значок с яблоком, резко повышающие стоимость компьютера. Вместо этого вы покупаете отдельные компоненты и получаете то, что вам нужно. Тщательный выбор позволит вам сэкономить сотни дензнаков. Самостоятельный выбор позволит также избежать второго компромисса — фиксированного оборудования, поскольку вы сами можете подбирать и компоновать детали, идеально подогнав их под свои цели. Вы собираете компьютер Linux для работы с документами, и у вас уже есть устройство NAS? Тогда вам ни к чему огромный жесткий диск на 4 TB. Если же вы рьяный фотограф, SSD небольшого объема будет для вас бессмысленным. То же можно сказать о видеокартах: встроенные видеокарты и от Intel, и от AMD ныне настолько мощные, что для повседневного использования вам, вероятно, не понадобится отдельная видеокарта. С другой стороны, если вы планируете заняться редактированием фото и видео, потребуется достойный GPU; а если вы собрались играть в самые современные игры, то нужно нечто еще более мощное. Для создания и запуска домашнего сервера подойдет маломощное, но надежное оборудование. А поскольку вы создаете машину с нуля, на ней не будет ненужных раздражающих операционных систем, и вы прекрасно установите выбранный вами дистрибутив.

Главный выбор

Ключевым здесь является слово «выбор». Одно из основных достоинств Linux, так же, как и свободного ПО и ПО с открытым кодом, заключается в огромном выборе. Он позволяет вам работать с операционной системой и программами, идеально соответствующими вашим потребностям — а если не идеально, то всегда можно поработать с исходным кодом и создать собственные, сугубо индивидуальные инструменты. Создавая собственный ПК, настраивая его и точно зная, как именно он работает, вы получите действительно персональный компьютер, а не просто компьютер «с прилавка». Невозможно переоценить важность и эффективность наличия как оборудования, так и программ, которые соответствуют вашим требованиям и останутся таковыми в течение нескольких лет.

Некоторые полагают, что сборка собственного компьютера — деяние исключительной сложности, но на самом деле это на диво просто. Например, у вас не получится присоединить к слоту не тот компонент по причине различия разъемов; и большинство материнских плат поставляются с инструкциями, очень ясными и простыми для понимания. Именно поэтому мы не стали включать пошагового руководства. Самый важный, и потенциально самый сложный, процесс в сборке собственного ПК — это стадия планирования, когда надо убедиться, что все части совместимы между собой и найдены подходящие для работы компоненты. Мы опишем вам все, что нужно для сборки компьютера, идеального для ваших задач, будь то стандартный настольный ПК, игровой ПК, центр мультимедиа или домашний сервер.

Собираем настольный ПК

Нужно что-то для обычного повседневного использования? Тогда вам сюда.

ПК многим нужен для простой повседневной работы. Если вы собираете именно такой компьютер, вам предстоит весьма деликатная задача поиска нужного баланса. Вам вряд ли потребуются дорогая и мощная видеокарта — как мы уже отметили, встроенные видеокарты в новейших процессорах Intel и AMD отлично подойдут для стандартной работы. Также не стоит устанавливать 16 ГБ самого скоростного ОЗУ. С одной стороны, желательно снизить расходы, установив минимум компонентов, но с другой стороны, стоит задуматься об обеспечении будущего своего ПК. Что мы имеем в виду? Ну, пока вам, возможно, хватит и 4 ГБ ОЗУ — возможно, даже с лихвой. Однако мы уже наблюдаем, как 8 ГБ ОЗУ на современных компьютерах становятся все более стандартными, и вряд ли вы захотите часто вскрывать корпус своего ПК для модернизации. Приобретение ОЗУ меньшего объема может оказаться вовсе не таким уж выгодным, как вы думали (более подробную информацию см. во врезке «Цена памяти»).



➤ Intel по-прежнему трудно превзойти, когда речь идет о лучших процессорах.



Приобретение современной материнской платы означает, что вам не придется переживать по поводу покупки определенных других компонентов. В прошлом мы бы порекомендовали купить звуковую карту; но практически все материнские платы в наши дни идут с хорошим встроенным звуком. То же можно сказать и о сетевых адаптерах Ethernet — больше незачем приобретать отдельную сетевую карту для повседневного использования, поскольку на вашей материнской плате уже есть встроенная. А значит, вам не придется изыскивать средства на добавочные компоненты.

Какой процессор?

Итак, что вам нужно купить? Первое, о чем стоит подумать — выбор процессора, поскольку это определит, какой материнской платой вы сможете пользоваться, из-за типа используемого разъема. В порядке отступления — все процессоры настольных ПК сейчас 64-битные (исключение составляют Intel Atom и более старые процессоры Intel Celeron), и вам понадобится именно такой, поскольку мы собираемся посоветовать вам взять не менее 4 ГБ памяти. Это ничуть не проблема, поскольку все дистрибутивы также предлагают 64-битные сборки.

Итак, памятуя, что вам нужен ПК средней мощности для повседневного использования, мы бы рекомендовали выбрать Intel Core i5-4570. Это очень мощный процессор, который стоит весьма недорого (его можно купить этак за £140 [для получения примерной стоимости в руб. умножьте на 60, — прим. ред.]), благодаря тому, что в нем нет продвинутых функций, присутствующих в более дорогих процессорах Intel. Например, в Intel Core i5-4570 заблокирован умножитель CPU; но поскольку вы не собираетесь разгонять процессор, чтобы не упустить ни грамма мощности, это не станет проблемой. В отличие от дорогих Intel, в нем четыре потока, а не восемь, и он не поддерживает многопоточность [Hyper-threading]. Как раз по данной причине это не самый лучший чип для профессионального редактирования видео и 3D-рендеринга — но для выполнения подобных задач вы вряд ли обойдетесь стандартным ПК. Он более чем достаточен для большинства задач, а встроенная графика (Intel HD Graphics 4600) отлично справится с мультимедиа, редактированием фото и даже с некоторыми играми.

Если вы ищете нечто покрупнее и у вас есть на это деньги, то отличным процессором будет Intel Core i7-4770K, хотя он немного дороже — около £242. С другой стороны, при желании сэкономить хорошим решением станет Intel Core i3-4130 — всего £83. Интересно, что у него тактовая частота выше, чем у Intel Core i5-4570 (3,4 ГГц по сравнению с 3,2), но он двоядерный, а i5 — четырехъядерный. Все эти процессоры используют разъем LGA 1150 (об этом чуть позже), что значительно облегчит поиск материнской платы.

Вероятно, вы заметили, что данный раздел нашей статьи перенасыщен Intel; если вы склоняетесь в сторону процессора AMD, отличной альтернативой будет A10-7850K. У него встроенная видеокарта AMD Radeon R7, и он стоит порядка £135, т.е. немного меньше, чем i5-4570. С этим процессором нужно будет искать материнскую плату, которая поддерживает сокет FM2+. Все упомянутые здесь процессоры должны идти с радиатором и вентилятором. Хотя они вас не сильно спасут, если вы — заядлый игроман или убежденный оверклокер,



Цена памяти

Для подавляющего большинства пользователей ПК 4 ГБ ОЗУ будет вполне достаточно для повседневной работы. Однако ОЗУ на 8 ГБ очень быстро становятся нормой для новых компьютеров, и по мере усложнения дистрибутивов и программ мы, возможно, скоро удостоверимся, что 4-ГБ ОЗУ сделались довольно-таки... старомодными.

Один из основных аргументов в пользу 8 ГБ против 4 ГБ — 8 ГБ ОЗУ гарантируют, что в ближайшее время ваш компьютер не устареет, тогда как при наличии 4-ГБ ОЗУ вам, вероятно, вскоре придется его обновлять. Мало того, что это мороза, но в конечном итоге может еще и встать

дороже. ОЗУ DDR3 уже изрядно зажились, и время DDR4 не за горами. Это означает, что цены на DDR3 очень сильно упали, и разница в цене между 4 ГБ и 8 ГБ вовсе не так велика, как можно по незнанию вообразить. Средняя цена 4-ГБ ОЗУ в настоящее время около £20–25, а 8 ГБ — около £35–40, так что стоимость гигабайта говорит в пользу большего объема. Если вы можете себе это позволить, стоит приобрести ОЗУ побольше, обеспечив будущее своей машины. Для стандартного, повседневного ПК мы бы рекомендовали HyperX XMP Blu Red Series на 8 ГБ, которую на момент написания можно найти онлайн за £53.



➤ Трудно не порекомендовать системную память на 8 ГБ — по причинам цены, производительности и обеспечения будущего.

но для обычного настольного применения подойдут отлично, и к тому же вам не придется покупать дополнительный радиатор или вентилятор.

Мамка

Определившись с процессором, пора задуматься о материнской плате. Она определит ряд важных аспектов собираемого ПК: например, размер корпуса и объем памяти, с которым она сможет работать. Если вы выбрали один из предложенных нами процессоров Intel, то материнская плата должна поддерживать чипсет Intel Z87 и разъем LGA 1150. Для стандартного ПК мы бы посоветовали материнскую плату ASRock Z87 Extreme3. Не пугайтесь слова “Extreme” в названии: на самом деле это весьма разумная плата за достойную цену. С ней вы получаете четыре слота DIMM для ОЗУ DDR3 (вплоть до солидных 32 ГБ), встроенный звук, четыре порта USB 3.0, два порта USB 2.0, встроенную сетевую карту и порты HDMI, VGA и DVI для подключения мониторов. Практически все, что вам нужно. Она идет с шестью портами SATA 3, так что вы сможете поставить несколько жестких дисков. Они имеют обратную совместимость с более старыми дисками SATA, и если у вас где-то завалилось несколько старых жестких дисков, можете их использовать — хотя если у них коннекторы IDE, не обойтись без адаптера. Если вам нужен новый жесткий диск, мы бы порекомендовали Western Digital Scorpio Black 500 ГБ.

Это достойный объем, а скорость у него стандартная — 7200 оборотов в минуту. Хотя твердотельные накопители быстрее, их относительно ограниченная емкость означает, что они вряд ли станут достойным приобретением для повседневной работы.

Чтобы обеспечивать энергией компоненты и сам компьютер, понадобится блок питания (PSU). Мы выбрали компоненты с относительно низким энергопотреблением, и поскольку у нас нет отдельной видеокарты, чудовищный блок питания в 1000 Вт не потребуется. Однако если в будущем вы планируете расширять свой ПК, неплохо будет купить PSU чуть мощнее, чем вам нужно сию минуту. Стоит также выбирать PSU понадежнее — вряд ли вы захотите, чтобы он сломался или повредил какие-то компоненты — и с хорошей эффективностью, поскольку ваш ПК будет постоянно использоваться. За самую дешевую модель хвататься не стоит, и прочтите сначала отзывы покупателей: это неплохой индикатор надежности блока питания. Что касается мощности, выбирайте блок питания, имеющий сертификат 80 Plus. Это означает, что он расходует не более 20% своей энергии на нагрев, сокращая и счета за электричество, и уровень шума.

Вам надо убедиться, что у блока питания подходящие для вашей материнской платы коннекторы. Все новые блоки питания с 2008 года соответствуют ATX12V v2.31, и если вы выбираете систему с блочно-модульными кабелями, это обеспечит его соответствие вашим потребностям, хотя обычно они используются только в дорогих моделях. Мы рекомендуем Corsair CX500, который объединяет надежность, мощность и эффективность всего за £46. По части корпуса не нужно ничего особенного. APEX SK-393-C Black Steel ATX Mid Tower Computer отлично подойдет, а на www.newegg.com он стоит всего £14.



➤ Блочно-модульная конструкция позволяет получить все нужные соединения питания, избежав путаницы кабелей.

Обычный ПК: Список покупок

Процессор	Intel Core i5-4570	£ 140 (www.ebuyer.com)
Материнская плата	ASRock Z87 Extreme3	£ 84 (www.kikatek.com)
Память	HyperX XMP Blu Red 8 ГБ	£ 53 (www.newegg.com)
Жесткий диск	WD Scorpio Black 500 ГБ	£ 45 (www.amazon.co.uk)
PSU	Corsair CX500M	£ 46 (www.amazon.co.uk)
Корпус	PEX SK-393-C Black Mid Tower	£ 14 (www.newegg.com)
DVD-привод	Samsung 24x SH-224DB DVD-RW	£ 11 (www.dabs.com)

Итого: £ 393 (~ 24 тыс. руб.)

Собираем игровой ПК

Наслаждайтесь приростом количества игр для Linux в своей системе.

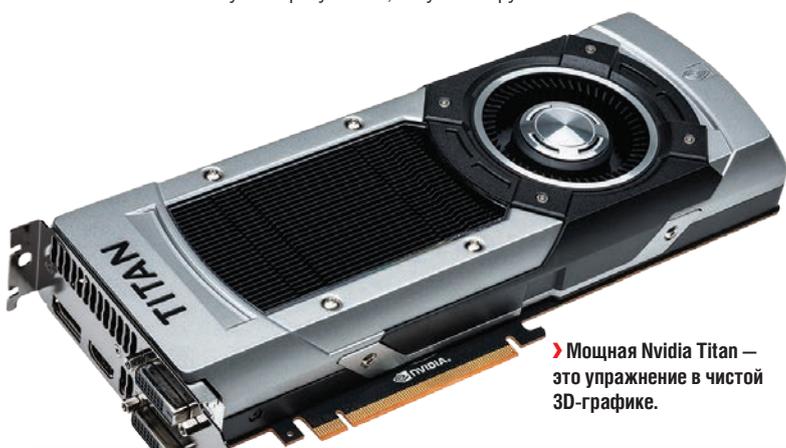
Всего несколько лет назад сама идея сборки игрового ПК, напигованного новейшими компонентами, с целью установки на него Linux привела бы вас к сожжению на костре или, как минимум, к покручиванию у виска вашими сотоварищами по играм. В наши дни все больше популярных блокбастеров игрового мира и инди-игр приходят в Linux, чему сильно способствует Valve — разработчик игр и компания, стоящая за Steam, сервером дистрибуции игр — которая создала собственный дистрибутив Linux, SteamOS. Благодаря тому, что компании вроде Crytek (серии *Crysis* и *Far Cry*) и Epic (*Gears of War* и *Unreal*), производящие популярные игровые движки, на которых работает множество крупных релизов — и производящие весьма впечатляющие в плане графики игры — полностью охватывают Linux, здоровье игр никогда еще не было настолько крепким.

Есть целый ряд причин, по которым вам может захотеться собрать игровую машину Linux. Для начала, Linux бесплатен, и вы можете потратить деньги, сэкономленные на лицензии Windows, на более мощную видеокарту — а это всегда очень популярная опция среди геймеров. Поскольку дистрибутивы Linux потребляют намного меньше ресурсов, работая в фоновом режиме, по сравнению с Windows, это также означает, что вы увидите куда лучшие результаты, запустив игру в Linux.

Так что же следует принять во внимание, собирая игровую машину Linux? Подход к созданию игрового компьютера, способного справиться с новейшими играми, обеспечив наилучшее возможное качество, сильно отличается от подхода к сборке стандартного ПК. Вам понадобится вся мощность, которую вы сможете позволить себе приобрести, а если вы затеете разгон процессора, то и компоненты, позволяющие выжать всю мощность до последней капельки. То есть вы должны быть готовы потратить несколько больше и иногда отказаться от дешевых компонентов ради более дорогих.

Планируем вашу машину

Вместе с тем, основа нашей игровой машины довольно сходна с основой обычного ПК. Материнская плата ASRock Z87 Extreme3, которую мы предложили для ПК, является отличным выбором, благодаря превосходной цене и поддержке ряда новейших технологий. В наших тестах она отлично показала себя в плане пропускной способности обращений к памяти и производительности разгона. Процессор Intel Core i5-4570, предложенный на предыдущей странице, тоже прекрасный выбор, хотя и не поддерживает разгон. Большинству его и хватит, но если вы хотите схитрить и получить за свои денежки дополнительную скорость, то Intel Core i5-4670K предлагает не самую быструю базовую тактовую частоту (3,4 ГГц по сравнению с 3,2 ГГц), но его также можно разогнать почти до 4,5 ГГц — а это солидный прирост совершенно бесплатно. Хотя Intel Core i5-4670K идет с охлаждающим вентилятором, для игрового ПК, в отличие от настольного, этого будет мало. Из-за интенсивности игр процессор при работе будет нагреваться намного сильнее, а особенно если его разогнать: тогда необходим другой кулер, чтобы обеспечить его нормальное состояние. Здесь есть две основные опции охлаждения: воздушное и водное. Воздушное обычно обеспечивается большими радиаторами и вентиляторами, расположенными над процессором. Достоинства воздушной системы охлаждения в том, что обычно она недорога и проста в установке, а недостатки — в том, что она довольно громоздкая и шумная.



➤ Мощная Nvidia Titan — это упражнение в чистой 3D-графике.

Составляющие производительности

Если вы хотите выжать из своего игрового ПК максимум производительности, вам понадобится самое быстрое оборудование, какое вы сможете позволить. Если говорить об ОЗУ, то его объем не так важен (8 ГБ по-прежнему хватит для игр), как частота и время задержки ОЗУ. Если вы решили использовать предложенную материнскую плату ASRock Z87 Extreme3, она поддерживает двухканальную ОЗУ DDR3 с четырьмя слотами, так что нам понадобится ОЗУ из двух или трех планок (например, две планки по 4 ГБ).

Частота ОЗУ может быть между 800 МГц и 3000 МГц; наша материнская плата при разгоне поддерживает частоту 2933. Для игр не требуется больше 1600 МГц. Чем быстрее ОЗУ, тем заметнее задержка, так что это неплохая уступка. Более быстрое ОЗУ к тому же сильнее греется, что,

в свою очередь, потребует наличия около планок памяти больших радиаторов. А это способно привести к проблемам, если у вас большой кулер CPU: вы можете обнаружить, что они не влезают в корпус. Corsair Vengeance Low Profile отвечает всем требованиям и имеет вполне посильную цену — где-то в районе £60.

Важнейшая часть игрового компьютера — видеокарта, и она же самая дорогая; однако стоит приплатить, чтобы обеспечить будущее своего компьютера. Самая мощная одиночная видеокарта на сегодняшнем рынке — Zotac GeForce GTX Titan Black, с огромным объемом памяти 6 ГБ GDDR5, 384-битным интерфейсом и пропускной способностью 3,36 ГБ/сек. Эта карта прослужит долго, и ее стоит купить, если гнаться за наилучшим. Однако она стоит около £800, чего многие не могут

себе позволить, и определенные аспекты карты, например, 6 ГБ GDDR5, не будут использоваться в полную силу даже самыми требовательными играми, особенно если используется только один монитор. Однако когда в норму выйдут дисплеи 4K, GDDR5 станет жизненно важна.

Для более реалистичных бюджетов мы рекомендуем AMD Radeon R9 290. И хотя от ее цены тоже не отмахнешься (около £320), она выигрывает по сравнению с картами Nvidia, которые дороже практически вдвое. Все эти мощные компоненты требуют сильного потока воздуха для охлаждения, и симпатичные большие шасси с достаточным числом встроенных вентиляторов отлично справятся с этой задачей; рекомендуем Corsair Carbide 200R, поскольку он очень даже вписывается в ценник — около £50.

Игровые процессоры

Для сборки игрового ПК мы снова обратились к процессорам Intel. Это не связано с какой-либо предвзятостью с нашей стороны (и под нашими столами не скопились мешки с деньгами); просто процессоры Intel, как правило, опережают предложения AMD, хотя AMD тоже начал набирать силу.

Нынешний урожай от Intel, 4-е поколение процессоров Intel Core, может похвастаться отличной скоростью — на радость игроманам. Ко времени вашего чтения должна поступить в продажу

новехонькая линейка процессоров IV поколения (кововое название — Каньон Дьявола [Devil's Canyon]), предлагающих прирост скорости относительно предыдущих процессоров Haswell при той же цене. Хотя процессоры «верхней» части линейки помечены i7, мы рекомендуем вариант i5. Дело в том, что процессоры i7 в основном отличаются от i5 поддержкой многопоточности — и справляются с большим количеством потоков (восемь против шести). Благодаря этому i7 отлично подходят для интенсивной

многозадачности, например, кодирования видео, но для игр тут пользы мало — и уж всяко недостаточно, чтобы оправдать переплату. Можно утверждать, что зато важно наличие в i7 большего кэша процессора (8 МБ против 6 МБ в i5), но при играх вы опять же особой разницы не обнаружите, и именно поэтому мы рекомендуем выбрать процессор i5 и поберечь свои денежки — или вложить их в более качественную графическую карту, что, безусловно, окажет больше влияния.

Системы водного охлаждения, с другой стороны, переносят тепло с процессора на радиатор, расположенный где-то на шасси компьютера. Преимущества водного охлаждения в том, что оно более эффективное и тихое, чем воздушное, и кроме того, может быть менее громоздким и более гибким. Недостатки — его сложнее установить, и оно несколько дороже. Однако можно купить готовую независимую систему водного охлаждения, с добавлением хладагента; тогда установка будет довольно простой. И хотя вас, возможно, не приводит в восторг мысль о протекании жидкости рядом с вашими дорогими электрическими компонентами, водные кулеры совершенно безопасны и надежны. Выбор будет зависеть от величины вашего корпуса и от того, насколько тихая работа ПК вам нужна и насколько дорогие компоненты позволяют вам ваш бюджет. Если вы не собираетесь заниматься разгоном и не против шума при работе ПК, хорошим выбором будет воздушное охлаждение; а если вам нужна тихая машина с большей степенью охлаждения, стоит остановиться на водном.

Вам также понадобится отдельная видеокарта — и ваш выбор зависит от того, насколько вам важно, чтобы на вашем игровом ПК самые современные и требовательные игры работали достойно. Вы можете заплатить за видеокарту всего £60, не рискуя потратить £1000, так что крепко подумайте, чего вы ждете от своего компьютера.

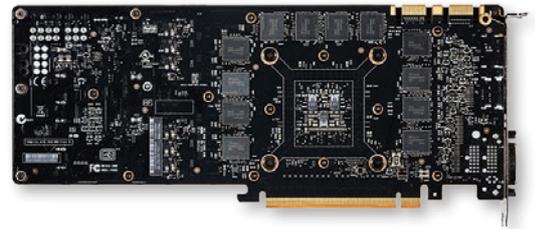
Драйверы и программы

Исторически так сложилось, что одним из главных камней преткновения, мешающих развитию игр в Linux, была необходимость заставить работать графические 3D драйверы. В лучшем случае драйверы были проприетарными, а в худшем их вообще не существовало. К счастью, всегда имелось преданное сообщество, которое откапывало альтернативы с открытым кодом, обычно содержащие те же функции, что и официальные драйверы £D.

Многие дистрибутивы облегчают выбор между проприетарными драйверами и драйверами с открытым кодом, в зависимости от ваших предпочтений. Например, в Ubuntu и его производных вы можете найти их, открыв System Settings, затем щелкнув по Software & Updates. Во вкладке Ubuntu Software щелкните по Proprietary Drivers for Devices, чтобы включить эти драйверы. Щелкните по вкладке Additional Drivers, чтобы посмотреть, какие драйверы установлены и есть ли среди них проприетарные.

Если вы используете карту Nvidia, то драйверами по умолчанию должны быть драйверы Nouveau с открытым кодом, созданные посредством декомпиляции проприетарных графических драйверов Nvidia для Linux. И хотя Nvidia далеко не любимчик мира открытого кода, она все же заслужила аплодисменты за свой недавний вклад в драйвер с открытым кодом. То есть появился шанс на большой паритет функций у драйверов с открытым кодом и проприетарных, что означает меньше причин использовать последние. Вы можете узнать больше о Nouveau на <http://nouveau.freedesktop.org>. Если вы используете видеокарту AMD, то драйверы с открытым кодом должны быть установлены по умолчанию. Проприетарные драйверы (известные как Catalyst/fglrx) можно установить через менеджер пакетов. Примечание: сначала обязательно удалите старые драйверы.

Что касается лучших дистрибутивов, хотя в новом ориентированном на игры дистрибутиве от Valve, SteamOS, делается многое, он пока что на очень ранней стадии развития, и мы опасаемся его рекомендовать, если вы хотите использовать свой ПК также и для повседневной работы. Однако в будущем к нему стоит присмотреться. А для наиболее совместимой и нормальной работы мы бы порекомендовали производные Debian, где наилучшим выбором будут Ubuntu или Mint.



➤ Проприетарные драйверы продолжают досаждают дистрибутивам GNU/Linux, и это будет продолжаться еще некоторое время.

Игровой ПК: Список покупок

Процессор	Intel Core i5-4670K	£ 161 (www.ebuyer.com)
Материнская плата	ASRock Z87 Extreme3	£ 84 (www.kikatek.com)
Память	Corsair Vengeance Low Profile 8 ГБ	£ 62 (www.scan.co.uk)
Видеокарта	AMD Radeon R9 290	£ 305 (www.ebuyer.com)
Жесткий диск	WD Scorpio Black 500 ГБ	£ 45 (www.amazon.co.uk)
PSU	Corsair CX600M	£ 53 (www.amazon.co.uk)
Корпус	Corsair Carbide 200R	£ 46 (www.amazon.co.uk)
Кулер CPU	CoolerMaster Seidon 120V	£ 35 (www.scan.co.uk)

Итого: £ 791 (~ 47 тыс. руб.)

Соберем медиа-центр

Стильный способ наслаждаться своим видео и музыкой в гостиной.

Если вам интересно собрать ПК, который ненавязчиво размещается в гостиной и обслуживает все виды мультимедиа, различаемые вашими глазами и ушами, вам будет приятно узнать, что в наши дни это возможно, и некоторые избранные дистрибутивы помогут вам создать превосходную машину для мультимедиа, с комфортом применимую с телевизором.

Сборка игрового ПК в основном подразумевает большой размер и мощь, а вот сборка достойного ПК для мультимедиа — дело куда более тонкое. Вам незачем мощные компоненты — по сути, иногда вы будете искать как раз противоположное, поскольку вы собираете небольшую по размеру машину, большую часть времени включенную, которая работает как можно тише, а значит, вам не нужно, чтобы она сильно грелась. Если вы не жаждете

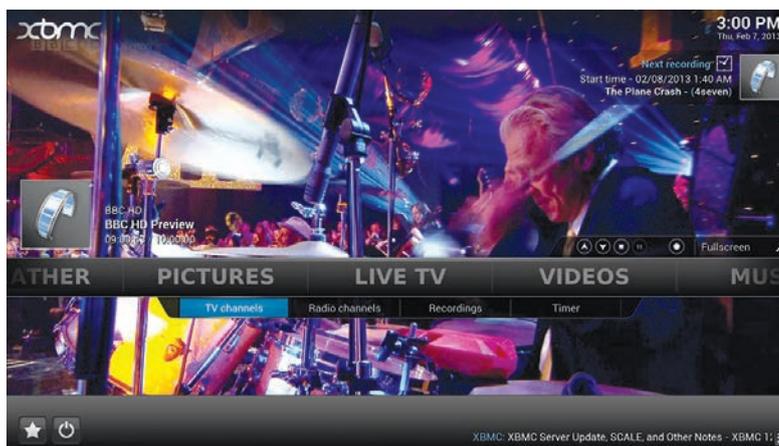
водрузить в углу комнаты огромный бежевый сундук, понадобится симпатичный и компактный корпус. К счастью, выбор велик благодаря форм-факторам малых размеров Micro-ATX — они стали очень популярны по мере обретения интегрированными компонентами все большей мощности и эффективности. Мы остановились на корпусе Antec ISK310-150 Micro-ATX, поскольку он компактный и отлично выглядит.

Из-за малого размера в нем довольно трудно ковыряться, и если вам надо больше простора и гибкости, мы бы также рекомендовали Antec ISK 600. Antec ISK310-150 Micro-ATX идет со встроенным 150-ваттным блоком питания, а стало быть, одной проблемой меньше. Скорость вентилятора на блоке питания тоже можно отрегулировать, чтобы он работал потише, хотя тогда придется приглядывать за температурой.

Идеально сформированные части

Из-за размера шасси вам нужно будет выбрать материнскую плату формы mini-ITX. Для данной сборки мы бы рекомендовали AS-Rock FM2A88X-ITX+: у нее не только компактный дизайн, но она также обеспечивает некоторые удобные функции для мультимедийного ПК. Она идет с выходом HDMI для подсоединения телевизора с высоким разрешением, да еще и входом HDMI для подсоединения других устройств, что особенно удобно, если у вашего телевизора число портов HDMI ограничено.

Материнская плата также идет со встроенными беспроводными антеннами 802.11n, что дает большую гибкость при размещении ПК. Она поддерживает сокет FM2+, подходящий для процессора AMD A10-7850K, рекомендуемого благодаря его превосходной встроенной графике, которая снижает и расходы, и энергопотребление. Для мультимедийного ПК идеальным решением по охлаждению процессора будет жидкостный кулер с относительно тихой работой, но в небольшом корпусе ему будет мало



➤ XBMC справляется с версией 13 Gotham, и это большая удача для всех.

Дистрибутивы для воспроизведения мультимедиа

В отличие от других систем, о которых мы рассказали и которые вы можете создать самостоятельно, ПО, устанавливаемое на мультимедийный ПК, не менее — если не более — важно, чем оборудование, на котором вы собираетесь его устанавливать. Есть ряд легковесных дистрибутивов, созданных специально для мультимедийных ПК. Их суть в том, чтобы обеспечить простую и одновременно элегантную операционную систему, которая предоставит вам простой доступ к вашим мультимедиа через интерфейс, специально разработанный для работы с телевизором.

Самое популярное ПО — XBMC (<http://xbmc.org>), некогда бывшее программой медиа-центра для консоли Xbox; но сейчас это полноценный медиа-плеер с открытым кодом, пригодный для установки на самых разных платформах. Дистрибутив идет с рядом отличных функций, в том числе с совместимостью с UPnP, которая позволяет воспроизводить мультимедиа с других подключенных в сеть устройств и на них, и с PVR, поддержкой

фильмов и музыки. Функциональность расширяется дополнениями. Поскольку XBMC устанавливается, как стандартный дистрибутив, вы сможете использовать свой мультимедийный ПК также и для повседневных задач. Однако если вы хотите, чтобы это был исключительно мультимедиа-плеер, стоит попробовать XBMCbuntu.

Как ясно из названия, это дистрибутив на базе Ubuntu, в котором уже установлен и настроен XBMC. Он разработан по возможности более легковесным, что позволяет максимально быстро загружать и смотреть мультимедиа, и идет в виде live CD, так что вы можете попробовать его перед установкой; более подробную информацию см. на <http://bit.ly/ixfbmcubuntu>.

Еще один популярный мультимедиа-центр — Plex, поставляемый в двух видах: Plex Home Theater можно устанавливать на компьютерах, соединенных с ТВ, и он умеет воспроизводить как локальные, так и удаленные мультимедиа; а другая версия — Plex media server — устанавливается на удаленные

компьютеры и устройства NAS, чтобы упорядочить и транслировать ваши мультимедиа на другие подключенные устройства. Хотя изначально это было ответвление XBMC, сейчас Plex обрел собственную жизнь, и его можно найти на многих Smart TV и приставках. Он использует ряд проприетарных технологий и технологий с закрытым кодом. Хотя Plex Home Theater не имеет на данный момент официальной версии Linux, есть сборки для определенных дистрибутивов, о которых вы сможете узнать на <http://bit.ly/ixfplex>.

А перспективный дистрибутив, который очень нравится нам в Linux Format — это OpenELEC (что означает Open Embedded Linux Entertainment Center). Это легковесный дистрибутив Linux, созданный с нуля исключительно для воспроизведения мультимедиа. Он также основан на XBMC и исключительно прост в настройке, и его можно установить практически на любое по конфигурации оборудование, включая, естественно, до смешного популярный Raspberry Pi.



Мультимедиа-центры на Raspberry Pi

Создание поюще-пляшущего мультимедийного ПК — это прекрасно, если вы хотите хранить и воспроизводить свои файлы мультимедиа локально, при этом транслируя контент с высоким разрешением и используя такие расширенные функции, как запись ТВ с помощью дополнения PVR (personal video recorder); но если вы уже храните свои файлы мультимедиа на другом устройстве, например, на диске NAS, и вам нужна машина всего лишь для трансляции контента на вашу домашнюю сеть, то вам ни к чему большая мощь.

На самом деле, Raspberry Pi является отличным устройством именно для этого. Он маленький и тихий, и хотя он не слишком мощный, ему вполне хватит энергии, чтобы транслировать мультимедиа, а встроенный порт HDMI делает его отличным выбором для работы с HDTV.

Еще лучше то, что очень многие из упомянутых нами дистрибутивов имеют версии, запускаемые

на Raspberry Pi: в частности, это OpenELEC. Нужно всего лишь скачать текущий релиз для Raspberry Pi с <http://bit.ly/lxfopenelecpi> и проверить наличие свободной карты памяти SD для установки на нее OpenELEC. После скачивания распакуйте файлы tar, затем в терминале с помощью команды `cd` перейдите в папку, куда вы только что распаковали файлы из tar-архива. Установив карточку памяти SD в свой ПК, обратите внимание на имя устройства (обычно нечто вроде `/dev/sdb1` — хотя на вашей машине это может быть не "sdb1"). Убедитесь, что вы определили устройство правильно, поскольку следующим шагом будет стереть все с выбранного вами диска. Теперь наберите `sudo ./create_sdcard /dev/xxx`, где `xxx` — это имя, данное вашей карточке памяти (например, `sdb1`). Затем наберите `sync` для завершения. Установите карточку памяти в свой Raspberry Pi и войдите в систему от имени пользователя root с паролем `openelec`.



► Raspberry Pi способен передавать HD-контент на ваш HDTV.

места. В таком случае Gelid Slim Hero будет неплохой воздушной альтернативой. Он достаточно мал, чтобы удобно разместиться над CPU, и, что самое важное, работает тихо.

Если вы собираетесь воспроизводить мультимедиа локально, вам понадобится большой жесткий диск для их хранения. Мы бы предложили обратить внимание на внешний жесткий диск-терабайтник Samsung M8 2.5" 9,5 мм SATA 6Gbps — это хороший выбор, поскольку он быстр и предлагает большие просторы для хранения. А так как это жесткий диск для ноутбука, он легко встанет в корпус малого размера — но перед покупкой проверьте совместимость жесткого диска и корпуса.

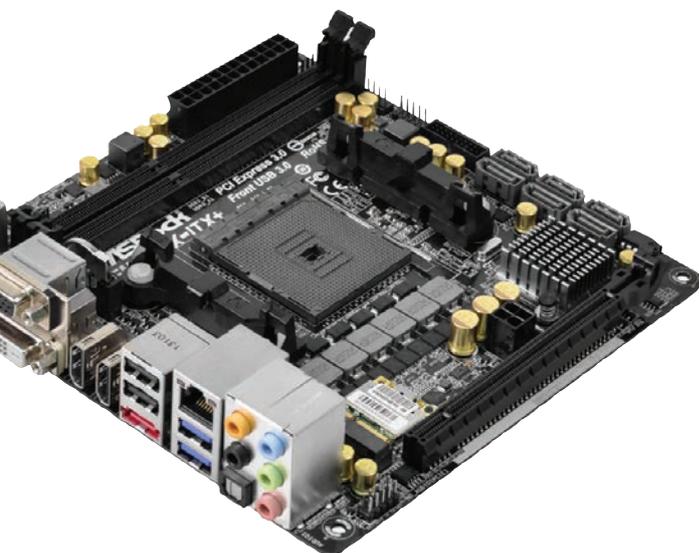
Кодеки мультимедиа

Чтобы воспроизводить мультимедиа на вашей новой системе, вам понадобятся всяческие кодеки. Они кодируют и декодируют цифровые файлы для воспроизведения, и различаются по форматам воспроизводимых файлов. Увы, большинство кодеков проприетарны и используют запатентованную технологию, обвешанную большим количеством лицензий, которыми управляет компания MPEG-LA, LLC.

MPEG-LA славится строгой защитой своих кодеков и быстрой реакцией в случаях, когда, по ее мнению, было допущено нарушение. Это привело ее к выступлению против Google, который развивал собственный проект WebM, предназначенный для использования кода VP8 в открытом и свободном медиа-формате. И хотя это только один случай, он ясно демонстрирует, что за кодеками часто стоит политика и разбирательства.

«Увы, большинство кодеков проприетарны и используют запатентованную технологию.»

Многие дистрибутивы дают возможность включить проприетарные кодеки для воспроизведения MP3-файлов и DVD или во время процесса установки, или позднее. Но что произойдет, если вы не хотите использовать проприетарные форматы? Есть ряд альтернатив с открытым кодом, в том числе OpenH264, Xvid и FLAC. Альтернатива для кодирования MP3, Lame, тоже доступна, хотя с ней вы ступаете на зыбкую почву, поскольку она применяет лицензированные технологии, не оплачивая лицензий. К сожалению, при сборке собственного мультимедийного ПК в некий момент вам придется выбирать между удобством и этическими проблемами, когда вы начнете думать о том, какие мультимедиа хотите воспроизводить. Мы надеемся, что ситуация скоро улучшится — даже MPEG-LA и Google сумели прийти к соглашению по поводу кода VP8, так что есть повод верить, что и другие кодеки последуют этому примеру.



► Эта платформа Micro-ITX является отличной основой для мини-мультимедиа-центра.

Медиа-центр: Список покупок

Процессор	AMD A10-7850K	£ 129 (www.ebuyer.com)
Материнская плата	ASRock FM2A88X-ITX+	£ 70 (www.lambda-tek.com)
Память	HyperX XMP Blu Red Series 8 ГБ	£ 39 (www.newegg.com)
Жесткий диск	Samsung M8 1 ТБ 2.5"	£ 55 (www.amazon.co.uk)
Корпус	Antec ISK310-150 Micro-ATX	£ 71 (www.ebuyer.com)
Кулер CPU	Gelid Slim Hero	£ 22 (www.quietpc.com)

Итого: £ 386 (~ 23 тыс. руб.)

Собираем домашний сервер

Обеспечьте целостность своих данных, которую вы заслуживаете, с помощью нашего сервера.

Собрать собственный домашний сервер? Интересное предложение — мы могли бы брякнуть какую-нибудь глупость, например, что можно запустить маломощный сервер на старом ноутбуке или другом позабытом металлоломе. Можно, конечно; но с точки зрения безопасности данных оно не слишком разумно. Это важный момент — вы должны спросить себя: зачем вам домашний сервер? Просто для прикола и ради эксперимента, обслуживания ваших персональных медиа-файлов дома и в Интернете, или для резервного копирования и распределенного доступа к файлам дома и в офисе? А может быть, все вместе, или еще по другим причинам?

Первые две задачи решаются и с помощью старого ненужного оборудования. Однако для всего, что требует хотя бы намек на сохранность данных, более достойной альтернативой будет взять за основу базовый настольный ПК (см. стр. 32) и усилить его хранилище полноценным решением RAID из четырех дисков.



➤ Чем больше жестких дисков, тем лучше для работы системы RAID.

А может быть, даже добавить UPS. Даже самый скромный корпус ПК должен позволить вам установить четыре диска, как с точки зрения соединения SATA, так и в плане блока питания и физических точек монтирования. Если с точками монтирования возникнут проблемы, можно купить наборы модернизации с 5,25 дюймов до 3,5 дюймов, которые позволят вам ввести и игру пространство в корпусе для оптических носителей. Но в принципе — кто в наше время пользуется оптическими дисками?

Чуть более скромным решением будет использовать два диска в зеркальной конфигурации RAID 1, однако вот что стоит принять во внимание, если вы вообще рассматриваете домашний сервер как способ хранения и резервного копирования ваших файлов: тут нужна надежная конфигурация, с избыточностью хранения данных, иначе за резервное копирование нечего и браться.

Поиск подходящего диска

Как мы видели, даже материнская плата относительно скромной ценовой категории предлагает достаточно портов SATA, чтобы поддержать программный RAID в Linux. Но какие диски стоит покупать? Как обычно, на сцену выходят три основных фактора: цена, производительность и надежность. Проблему надежности отчасти решает применение RAID; однако отчет BlackBlaze по жестким дискам (<http://bit.ly/LXFdrives>) уверенно называет самыми надежными диски Hitachi Deskstar 7K(1/2/3)000. За ними следуют диски Western Digital Red 3TB (WD30EFRX), а на третьем месте стоят диски Seagate.

Правда, отчет практически бессмыслен, поскольку подразделение жестких дисков Hitachi было куплено Western Digital, и на рынке остались диски WD вместе с Seagate и Toshiba. Что касается производительности, большинство 3,5-дюймовых дисков на 7200 об./мин. в этой категории равноценны. Цены колеблются, но в общем самым конкурентоспособным является Seagate, хотя и с очень незначительным отрывом.

Это RAID

Redundant Array of Individual Disks или RAID — это отказоустойчивая система, которая позволяет создавать системы хранения, способные выдерживать сбой дисков. Есть множество аппаратных контроллеров дисков RAID, однако в Linux вы также можете создать массивы RAID программно, используя стандартное оборудование, и для большинства случаев домашнего применения этого вполне достаточно.

Если вы никогда не сталкивались с отказом жесткого диска, значит, вы или очень молоды, или очень везучи. Диски отнюдь не вечные. Обычно производители заявляют среднее время между отказами до миллиона часов, но это среднее для всех дисков, то есть один из них на самом деле способен сбиться ежечасно, или годовой уровень отказов составляет 1%.

Правда в том, что реальный процент может быть намного выше. Данные Google Failure Trends in a Large Disk Drive Population [Тенденции отказов

в больших массивах жестких дисков] (http://static.googleusercontent.com/media/research.google.com/ru/archive/disk_failures.pdf) называют цифру между 2% и 6%, в зависимости от возраста диска. Если вы работаете с пятью дисками — что вполне реально — то, по словам нашего научного редактора, имеющего ученую степень в математических науках, есть 26% вероятности, что в год будет погибать один диск, хороня вместе с собой все свои/ваши данные. Если только нет резервных копий где-то еще.

Изначально RAID предназначался для избыточности при копировании целого массива дисков. В основном RAID 1 зеркально отображает равное количество дисков. Это довольно расточительно в смысле хранения — 50% потери в объеме — но работает с двумя жесткими дисками и требует минимальных ресурсов для обработки.

Веселье начинается с RAID 5: он распределяет данные по нескольким дискам, добавляя

контрольный разряд четности. Если один из дисков откажет, то можно восстановить весь массив. Дополнительное преимущество в том, что теряется меньше места; на самом деле, добавление большего количества дисков сокращает эту потерю. Он требует не менее трех дисков (с 30% потерей объема), и может быть расширен. При наличии пяти дисков потеря сводится к 20%.

Процесс реорганизации RAID 5 занимает немало времени, и если вы решите использовать более емкие диски на 1 ТБ+, данный промежуток времени может вызвать настоящую проблему, поскольку открывает массив для потенциального отказа. Проблема настолько сурова, что Dell в 2012 году решил рекомендовать отказаться от RAID 5.

К счастью, RAID 6 удвоил контрольный бит четности, позволив массиву справляться с отказом двух дисков, и это делает его привлекательным для больших массивов и дисков, где вероятность двойного отказа дисков повышается.



➤ Двойной NIC обеспечивает избыточность на случай отказа порта.



Наш материал посвящен сборке наилучшего компьютера для определенных задач; ну, а если такой уже где-то собран? Да. Дешевые целевые домашние серверы производит Hewlett Packard и продает под той же маркой HP ProLiant, что и его большие, взрослые рабочие станции. Суть в том, что все в сервере, от защелки на крышке корпуса до материнской платы, разработано, реализовано и собрано командой сервера HP. HP ProLiant Microserver — небольшой отдельный сервер всего 30 см шириной и 40 см в высоту и глубину, однако в нем умещаются четыре жестких диска. Спецификация основана на маломощном процессоре AMD Turion II Neo N54L и предлагает 4 ГБ памяти. Материнская плата, процессор и кулер предустановлены и включены в стоимость. Эта модель также обычно идет с жестким диском 250 ГБ, а если подсуесться, вы нарвете скидку £ 100. Так что за £ 185 у вас будет готовый сервер с местом для четырех дисков (пяти, если конвертируете место для оптического привода).

Столь много приятностей вы просто больше нигде не найдете. Например: точка монтирования для шестигранного инструмента внутри запираемой дверцы. Быстросъемные секции для крепления жестких дисков с легко монтируемыми съемными лотками плюс выдвижной лоток для материнской платы, и все это с быстроразъемными кабелями данных. Он не идеален, в нем всего два (труднодоступных) слота расширений, и работает только один порт Gigabit LAN, а на своих 6 портах он ограничен USB 2.0. Но это приемлемый компромисс для дома или небольшого офиса, и мы без проблем работали с таким 7/24 целых два года.

У всех есть резерв

Linux поддерживает RAID на уровне модуля отображения устройств в ядре, и настраивается с помощью `mdadm`, что позволит создавать, увеличивать и управлять устройствами с программным RAID. Для установки этого пакета наберите `sudo apt-get install mdadm` (в дистрибутивах на базе Debian), и вы сможете создать RAID относительно просто. Скажем, у вас четыре диска (`sdb`, `sdc`, `sdd` и `sde`), и вы бы хотели создать массив RAID 5; тогда введите `mdadm -C /dev/md0 -n4 /dev/sdb /dev/sdc /dev/sdd /dev/sde -l5` и задействуйте его командой `mdadm -D /dev/md0`.

Файловую систему на устройстве RAID можно создать командой `mke2fs -j /dev/md0`, и смонтировать ее обычным способом. Последняя проблема, которую нужно устранить — ваше свежесозданное новое устройство `/dev/md0` не будет снова собрано при перезагрузке системы, если только вы не добавите для этого строку в `/etc/mdadm.conf`. Чтобы добавить строку через `mdadm`, используйте

```
mdadm --examine --scan
>> /etc/mdadm.conf
```

Суть RAID — обезопасить себя на случай отказа жесткого диска. То же, однако, касается сетевых портов и питания. В домашних условиях отказ сетевого порта, скорее всего, наименее трудоемок. С этим мы сталкивались чаще, чем ожидали, при отказах и встроенных портов Ethernet, и добавочных карт. Они раздражают, и их сложно диагностировать, потому что обычно вы объясняете проблему отказом другой части сети, а не сетевой карты.

Решением будет работа с двумя портами Ethernet, что допускается большинством протоколов, таких, как Link Aggregation, Link Bundling, Port Trunking, NIC Bonding, NIC Teaming, LACP и наш любимый IEEE 802.1ax, ранее IEEE 802.1ac. Определенная настройка требует совместимых коммутаторов, но обычно балансирование нагрузки, круговая работа и active-backup должны работать с любым из них. Настройка не является темой данной статьи, но руководство для Ubuntu можно найти здесь: <http://bit.ly/LXFbonding>.

Итак, вам нужно `sudo apt-get install ifenslave-2.6`, затем `sudo stop networking`, и добавьте связующий модуль — `sudo modprobe bonding`. Сложность будет представлять правильное редактирование файла `/etc/network/interfaces` для вашей установки перед перезапуском сети с помощью `sudo start networking`. Редактирование мы отложим до другого раза, но должны отметить, что, несмотря на уверения Интернета, Link Aggregation не повышает скорость работы сети.

И последний пункт, которым вы, возможно, озаботитесь: это бесперебойное питание. UPS дает серверу время, достаточное, чтобы выключиться элегантно, идеально завершив все выполняемые задачи. Домашний UPS — всего лишь управляемая с ПК свинцово-кислотная батарея в корпусе: ПК через последовательный порт сообщает ей об обрыве питания. Батарея даст те самые жизненно необходимые пять–десять минут дополнительного времени работы для безопасного завершения. Простейшие варианты стоят умеренные £ 80; хотя для домашнего использования это, пожалуй, все-таки перестраховка. **LXF**

Учат в школе

По случаю того, что Raspberry Pi пошёл второй год, мы отправили Леса Паундера на Raspberry Jamboree, чтобы он поискал самый крутой Pi-проект.



Raspberry Pi был изначально предназначен для помощи детям в освоении компьютеров, и эта цель постепенно находит своё воплощение благодаря образовательному сообществу. Педагоги из всех уголков Соединённого Королевства приняли участие во второй выставке Raspberry Jamboree, впервые проведённой в рамках более крупного мероприятия Education Innovation Conference. Множество участников представили свои экспонаты,

и особенно радостно было наблюдать, что количество стендов с продуктами на базе Raspberry Pi не уступало самым именитым в сфере образовательных брендам.

Самопровозглашённый 'Jambassador' Алан О'Донохоу [Alan O'Donohoe] вёл мероприятие со свойственным ему напором и бешеным темпом презентаций. Алан невероятно много сделал для того, чтобы эта выставка стала такой, какая она есть, и я рад сообщить, что в 2014-м всё прошло «на ура». В этом году стало очевидно, что основной акцент теперь сместился на обучающий

потенциал Raspberry Pi, и эта идея была подхвачена сотнями учителей-участников. Сквозной темой множества практикумов и презентаций стало обсуждение вариантов использования этого экономичного компьютера на уроках. Выставка продлилась три дня, из которых первые два были посвящены обучению, а третий прошёл в форме открытой мастерской, где каждый мог представить собственный проект и узнать для себя что-то новое. Прекрасных идей было хоть отбавляй, и мы решили побеседовать с теми, кому они пришли в голову.



Мартин О'Хэнлон

Мартин О'Хэнлон [Martin O'Hanlon] — страстный фанат Raspberry Pi. Он любит Pi за способность увлекать детей программированием и более всего значит, пожалуй, своими толковыми и забавными проектами *Minecraft* с использованием API (о проектах на базе *Minecraft* см. на стр. 86, 88).

Linux Format: Вы известны как автор двух крутых проектов, но хотелось бы поговорить с вами о создании прекрасных уроков о *Minecraft*.

Мартин О'Хэнлон: Проекты *Minecraft* просты, компактны и задуманы как доступные. Поэтому, *Minecraft* — прекрасный способ «испытать себя» и поиграть с этим миром, из которого можно вылепить что угодно.

LXF: Вы создали классные вещи для *Minecraft* на Raspberry Pi. Почему вы предпочли именно эту платформу?

MX: По двум причинам. Во-первых, она бесплатна, что делает её доступной многим. Во-вторых, сам API (Application Programming Interface), используемый для создания мира *Minecraft*, очень прост в освоении и управлении. Программировать в *Minecraft* можно и другими способами, на других платформах, но ничто не сравнится с гибкостью и прозрачностью версии Raspberry Pi.

LXF: И что можно за относительно краткое время сделать в *Minecraft*, применяя API?

MX: Довольно быстро можно сделать четыре вещи. Заменить блок, определить, где он находится,



найти, где ваш игрок, и наконец, переместить игрока. Для этого в *Minecraft* применяются координаты хуз. То есть, чтобы ромбовидный блок оказался 10-м блоком в воздухе, вы можете с помощью API найти свой персонаж, добавить 10 по оси у и создать там блок. С одиночным блоком это забавно, но стоит вам освоить циклы в Python, и вы откроете способ создавать кучу вещей за секунды. Эти 4 вещи очень простые, но и очень мощные. Кроме того, *Minecraft* делает программирование интереснее для детей, поскольку мир *Minecraft* и его устройство им уже хорошо знакомы.

LXF: Какие же проекты были созданы с применением API *Minecraft*?

МН: Я видел множество игр на базе *Minecraft* — один парнишка, Николас Харрис [Nicholas Harris], в свои 12 лет уже написал несколько таких сам — и это круто!

LXF: Из вашего блога мы узнали, что вы смоделировали в *Minecraft* контуры Манхэттена — как вам это удалось?

МН: Это мой любимый проект. Он зрелищно описывает широту возможностей. И я его люблю не только за контуры Манхэттена, но и потому, что в основе его — стандартная 3D-модель, известная как OBJ-файл и применяемая во многих приложениях 3D-моделирования.

Формат OBJ используется при переносе работы из одного приложения в другое и на самом деле представляет собой просто набор координат и вершин, используемых при построении объекта. Этот OBJ-файл импортируется в *Minecraft* с помощью написанной мною программы, которая считывает эти массивные текстовые файлы и преобразует в данные для использования в *Minecraft*.

LXF: Если кто-то заинтересовался и хочет узнать больше, где им получить дополнительную информацию?

МН: Я разместил целую серию инструкций на моём сайте <http://stuffaboutcode.com>, так что любой может открыть его и попытаться себя в *Minecraft*. Как код, так и инструкции с этого сайта распространяются бесплатно, и я буду очень рад услышать о новых проектах.

О ВИДЕОРЕГИСТРАТОРАХ

«Камера на базе Raspberry Pi записывает мои выкрутасы на сноуборде.»

LXF: Не могли бы вы рассказать немного и о втором вашем потрясающем проекте?

МН: По жизни у меня три главных интереса. Технологии, автомобили и сноуборд. И я постарался внедрить Raspberry Pi в каждый из них. Сначала мне пришлось в голову использовать Raspberry Pi и камеру для записи моих поездок, потом я расширил эту идею, решив добавить данные на видео. Сюда входила скорость, число оборотов в минуту, температура радиатора и положение педали газа. Потом я решил перенести это на моё увлечение сноубордингом.

LXF: То есть у вас есть видеорегистратор для сноубординга...

МН: Во-во, это камера на базе Raspberry Pi, которая записывает мои выкрутасы на доске от первого лица и добавляя данные о температуре и высоте, плюс карту моих маршрутов. Это явно

более крутое решение, чем какая-нибудь GoPro камера.

LXF: Фантастика! Легко ли было это воплотить?

МН: Я взял за основу свой старый автомобильный видеорегистратор, так что большая доля работы была уже сделана. Корпус я изготовил из двух пластиковых пищевых контейнеров. В один уложил Raspberry Pi и батарейку, а в другой — устройство GPS. Пришлось слегка изменить проводку — в основном потому, что я не очень дружу с паяльником. А ещё я переписал код на Python, подстроив прибор под новые нужды. При тестировании обнаружился интересный сбой между моим GPS-устройством и официальной камерой Pi. Включённая камера конфликтовала с GPS — либо останавливая запись, либо искажая поток GPS-данных до неузнаваемости. Дело в том, что из-за высокой скорости передачи данных возникало электромагнитное поле. Решением стало обернуть кабель камеры фольгой: получается эффективное экранирование. Затем я его усовершенствовал, обернув кабель полиэфирным чехлом, чтобы аккуратнее смотрелось.

LXF: Какое ПО вы использовали?

МН: Всё было написано на Python, снабдившем меня богатейшими библиотеками. Для работы с GPS я использовал одну библиотеку, для наложения графики — другую, которой, естественно, была PIL (Python Imaging Library). На кадр графика накладывалась с регулярным интервалом, что создавало иллюзию её постоянного присутствия на экране.

LXF: А как именно осуществляется взаимодействие с Raspberry Pi? Простота интерфейса,

видимо, плюс, когда устройство связано с активным хобби?

МН: На время катания на сноуборде я надеваю толстые перчатки, которые прекрасно защищают меня от мороза, но неудобны при

работе с мелкими кнопками и переключателями. Эту проблему я решил, добавив большую красную кнопку — она осуществляет все управление. Я нажимаю её и для запуска, и для остановки записи. Если я хочу отключить Raspberry Pi, удерживаю нажатие несколько секунд.

LXF: Ваша камера на основе Raspberry Pi охватывает столько областей: и GPS, и видеозапись, и фиксация температуры и скорости. Что вы ещё планируете туда добавить?

МН: Моя очередная идея — добавить на дисплей биометрические данные. Мне бы очень хотелось записать своё сердцебиение и отобразить его на экране в виде графика. Это была бы фантастика. Мой проект видится мне лишь первой ступенькой в производстве видео с данными на Raspberry Pi, и мне не терпится узнать, до чего дойдут другие.



Эндрю Малхолланд

Эндрю Малхолланду [Andrew Mulholland] 17 лет, но репутация у него уже солидная. Титулованный изобретатель, робототехник и энтузиаст Raspberry Pi. Будучи самым юным в Северной Ирландии представителем STEM [сеть научных лабораторий Science, Technology, Engineering, Mathematics, — прим. пер.], он руководит множеством компьютерных клубов и мастерских.

Linux Format: Не могли бы вы побольше рассказать о своём проекте?

Эндрю Малхолланд: Мой проект — перенести Linux Terminal Server Project (LTSP) на Raspberry Pi. Для тех, кто об этом не слышал, суть такая, что имеется один центральный сервер, отвечающий на запросы клиентов в сети — клиентами выступают Raspberry Pi. Клиенты загружают минимум установленных программ, а затем ОС

и связанное с ней ПО по сети, через центральный сервер.

LXF: В Raspberry Pi по умолчанию отсутствует функциональность PXE (Preboot Execution Environment), как же вы заставляете их загружаться по сети?

ЭМ: Я сделал загрузочный образ на 27 МБ для SD-карты Raspberry Pi, в составе которого образ ядра и загрузчик. Вы, наверное, знаете, что Raspberry Pi может загружаться только с SD-карты, вот и надо разместить там эти файлы, чтобы создать «толстый клиент». Ещё там находится файл настройки, указывающий Pi тот IP-адрес, откуда запрашивается ОС через сеть.

LXF: Умно! Сколько же времени требуют, скажем, 4 или 5 штук Raspberry Pi на загрузку и переход в рабочее состояние?

ЭМ: Ну, это зависит от скорости сети. Хотя Raspberry Pi поддерживает соединение в 100 Мбит, лучше всего использовать гигабитное подключение на сервере, поскольку обслуживание пяти 100-Мбитных машин требует большей полосы пропускания. Я тестировал 20 Raspberry Pi, подключённых через гигабитную сеть, и все 20 машин были готовы к работе за 70 секунд.

ОБ LTSP

«Весь класс может хранить свои работы централизованно.»

LXF: Какие ещё преимущества даёт использование Linux Terminal Server Project на Raspberry Pi?

ЭМ: Главных преимуществ у LTSP два. Во-первых, там есть графический экран входа в систему.

Во-вторых, пользователь может войти в любой Raspberry Pi, не теряя доступ к своей домашней директории, как будто они находятся внутри одной системы. Поэтому весь класс может хранить свои работы централизованно, в одном месте, а не на разных SD-картах.

LXF: Какова же целевая аудитория проекта?

ЭМ: В основном те, кто связан с образованием. Главной целью было внедрить LTSP в учебный процесс. Плюс LTSP в том, что легко обеспечить наличие в вашем школьном компьютерном классе свежего образа Raspbian, и от этих хрупких SD-карт можно по большей части отказаться. Ученики могут сохранять работы в своих домашних папках и быть уверены, что ничего не потеряется.

LXF: А какого типа сервер потребуется для целого класса?

ЭМ: Не обязательно мощный. Это же просто файловый сервер. Единственно — я бы порекомендовал использовать гигабитный сетевой интерфейс. У меня прекрасно получалось запустить сеть с ноутбука на Debian с 1 ГБ ОЗУ.

LXF: Ваш проект подходит только для Raspbian? Можно ли его использовать с другими ОС для Pi?

ЭМ: Его можно использовать с другими ОС, если знать, как сделать образ. LTSP создавалась для работы с Ubuntu; моя программа как раз и упрощает создание загрузочного образа. Правда, сама компания внесла кучу изменений в исходный образ armhf, и это усложняет задачу — но не делает её невозможной. Чтобы создать образ самим, нужно понять, какие именно изменения были внесены, и добавить их в образ.





Саймон Уолтерс

Саймон Уолтерс [Simon Walters] обучает азам компьютерной грамотности младших школьников, о чём и пишет в своём блоге <http://cymplecy.wordpress.com> (произносится «симплиси»).

Linux Format: Расскажите, пожалуйста, о вашем проекте.

Саймон Уолтерс: Мой проект — версия Scratch с использованием GPIO (General Purpose Input/Output). Когда Raspberry Pi только появился, было во всеуслышание заявлено, что в его операционной системе, Raspbian, будет предустановлен Scratch. Широко освещалось и то, что в Raspberry Pi будет этот чудесный разъем GPIO, позволяющий работать с электронными компонентами.

LXF: И как вам пришла идея соединить Scratch и GPIO?

СУ: Ну, ещё на раннем этапе мне посчастливилось увидеть работу со Scratch Эндрю Робинсона [Andrew Robinson] и команды PiFace, а также их первую плату PiFace. Я связался с ними, и они любезно предоставили мне копию, а я её затем модифицировал для работы напрямую с GPIO. Я работаю в начальной школе, и хотел сделать Raspberry Pi и программирование как можно проще, так что выбор Scratch был естественным.

LXF: Что же умеет ScratchGPIO?

СУ: На простейшем уровне, он может включать и выключать светодиод или считывать статус переключателя. Ещё он может управлять моторами, а освоив этот аспект, вы окажетесь уже недалеко от создания своего первого робота.

LXF: Для Pi робототехника — родная сфера; перспективен ли здесь Scratch как язык?

СУ: Да, весь необходимый код для создания своего первого робота написать легко. Моё ПО для Scratch GPIO работает «за кадром» — пользователи его не видят. Они программируют робота через интерфейс, и он просто работает.

LXF: Насколько ScratchGPIO сопоставим, скажем, с использованием Python с GPIO?

СУ: ScratchGPIO использует кодовую базу PiFace и библиотеку Python RPi.GPIO Бена Кростона [Ben Croston], так что функциональность ScratchGPIO зависит от Python.

LXF: Значит, базируя свой проект на уже существующих, вы сразу получаете доступ к основной функциональности?

СУ: Совершенно верно — как только кто-то напишет код Python для работы, скажем, с ультразвуковыми датчиками, я тут же адаптирую его в ScratchGPIO и упрощу работу с ним.

LXF: А сам процесс переноса кода из Python в Scratch состоит просто в чтении его структуры и последующей адаптации?

СУ: Да. Например, если вы написали код на Python, позволяющий менять скорость работы мотора, вам достаточно создать переменную

запусков. Это прекрасное применение ScratchGPIO, но не стоит забывать, что основное его назначение — увлечь младших школьников изучением электроники с помощью простых проектов, наподобие светофоров и сенсоров.

LXF: Как раз такой вы продемонстрировали около года назад на Raspberry Jam в Манчестере — робота, умеющего проходить лабиринт. Во сколько обошёлся подобный проект?

СУ: Тот проект был на редкость дешёв: я обнаружил, что можно прикрепить шаговые двигатели к корпусу Raspberry Pi обычным двусторонним скотчем. Все обошлось в £10.

LXF: ScratchGPIO — один из основных языков, поддерживаемых последней разработкой Piromoni и Cyntech — съёмной платой Pibrella. Что она собой представляет?

СУ: Pibrella — готовая приставка, позволяющая сразу взяться за порты ввода и вывода. Не требуется ни навыков работы с паяльником, ни покупки дополнительных плат и компонентов. Все уже есть на ней самой.

LXF: Значит, Pibrella — только для новичков?

СУ: Нет, она растёт вместе с уровнем знаний владельца. Вы начинаете со встроенного светодиода и кнопки. А изучив электронику глубже, вы станете подсоединять к Pibrella сенсоры, переключатели и моторы.

LXF: То есть Pibrella годится и для управления моторами в робототехнике?

СУ: На базовом уровне — да. С её помощью довольно легко сделать двухмоторного трёхколёсного робота, и одно из колёс будет рулевым. За £10, Pibrella — прекрасное приобретение. У неё куча функций, причём они подстраиваются под нужды пользователя. **LXF**

О PIBRELLA

«С её помощью легко сделать двухмоторного трёхколёсного робота.»

в Scratch — с именем motorA — которая будет отслеживать изменения. Когда значение переменной в Scratch меняется, сигнал об этом передаётся по локальной сети, и программа-получатель следит за этими изменениями, а ScratchGPIO их обрабатывает.

LXF: Какие проекты были сделаны с использованием ScratchGPIO?

СУ: На ScratchGPIO сделано множество роботов, но меня заинтриговал проект человека, который хотел запускать с его помощью фейерверки. GPIO использовался как интерфейс механизма запуска петард, а Scratch — для управления серией



Третий вариант

Игорь Штомпель поговорил с Владимиром Рубановым, президентом компании «РОСА», о проблеме выбора.



Владимир Рубанов, президент и генеральный конструктор компании «РОСА», выступая на Московском деловом форуме 2014 года в секции «IT-технологии для

государства и бизнеса», обозначил три подхода к выбору программного обеспечения с точки зрения страны: первый — использование зарубежного программного обеспечения; второй — создание отечественного ПО «с нуля» российскими специалистами; третий — создание отечественного ПО на базе комбинации открытых и свободных международных компонентов с собственными разработками (<http://www.rosalab.ru/blogs/svobodnoe-po-kak-tramplin-dlya-tehnologi>). Чтобы прояснить детали, мы попросили В. Рубанова ответить на наши вопросы.

Linux Format: Чем вы занимаетесь в компании «РОСА»?

Владимир Рубанов: Расшифровка «РОСА» содержит сочетание «Российские операционные системы». Соответственно, главная область специализации компании — отечественное системное ПО, в том числе сертифицированное по требованиям защиты информации. Мы сотрудничаем с производителями аппаратного обеспечения и с производителями прикладного ПО, совместно обеспечивая решения для автоматизированных систем заказчиков.

LXF: Какие подходы к выбору ПО для обеспечения безопасности и независимости страны вы видите?

ВР: Я бы выделил два аспекта — экономический и политический. С точки зрения макроэкономики, чем больше отечественных товаров и услуг потребляется внутри страны, тем лучше стране — это и рабочие места, и развитие отечественной отрасли, и много других хороших эффектов. Поэтому импортозамещение ПО и связанных с этим услуг стране полезно по сравнению с чистым переводом средств за рубеж (импорт лицензий). Однако форсировать такое импортозамещение для всех категорий потребителей не стоит. Для технологической независимости страны достаточно обеспечить независимость от зарубежного ПО в критических областях, а затем уже постепенно развивать высокотехнологическую отрасль по рыночным механизмам, в том числе сотрудничая со всем миром.

LXF: Что можно сказать, в данном контексте, об использовании зарубежного ПО? В чем плюсы, а в чем заключаются минусы?

ВР: Главный плюс в том, что зарубежное ПО уже существует и работает. Это дает самый легкий путь — плати за лицензии и используй готовое. Однако минусы имеются — от макроэкономических эффектов из-за чистого импорта лицензий, когда деньги уходят на поддержку чужой экономики, до рисков национальной безопасности — зарубежные производители могут в любой момент из-за политической воли руководства своих стран прекратить поддержку ПО и отозвать лицензии. Угрозу силовых сценариев с активацией каналов утечки информации и удаленного нарушения функционирования через закладки тоже никто не отменял.

LXF: Какие угрозы оно несет в себе? Известны ли вам примеры реализации этих угроз?

ВР: Многое на тему использования ПО для шпионажа и закладок для потенциальных атак стало в публичной плоскости известно благодаря Эд-

для создания на его основе отечественных продуктов силами российских компаний.

LXF: Расскажите о третьем подходе, основанном на комбинировании отечественных разработок со свободным и открытым ПО.

ВР: Этот подход позволяет сочетать синергию работы с международным сообществом с созданием контролируемых компанией версий продукта. С точки зрения ресурсов — грубо говоря, мы берем 90% открытого кода, добавляем 10% своих фирменных разработок, все это интегрируем, тестируем и выпускаем составной продукт, за который можем отвечать как отечественный производитель. При этом исходные коды на 100% продукта, средства его сборки и открытые лицензии обеспечивают юридическую и техническую возможность иметь полный контроль над продуктом. Остается добавить компетентных специалистов, которые в этом коде разбираются в достаточной для поддержки степени. Таких специалистов нужно гораздо меньше по сравнению с разработкой всего «с нуля».

Для регулярного обновления версий полезно выстраивать своего рода конвейер [pipeline]: сначала делается бесплатная публичная версия; после обкатки на ее основе создается корпоративная; наконец, завершают цепочку защищенные сертифицированные варианты, в которых к выверенной корпоратив-

ной версии добавлены соответствующие средства защиты информации. Безусловно, по мере работы над таким конвейером важно по возможности максимум выполненных исправлений и доработок передавать обратно мировому сообществу [upstream]. Это обеспечивает как сокращение затрат при переходе на новые версии, так и увеличение авторитета и степени влияния на базовую разработку.

LXF: В контексте третьего подхода, что можно сказать о решениях, предлагаемых компанией «РОСА»?

ВР: Производственный конвейер компании «РОСА» выстроен именно по принципу «от бесплатной открытой версии до сертифицированных защищенных вариантов». Самое свежее ПО можно найти в бесплатных версиях ROSA Fresh; на основе стабилизации и доработки Fresh готовятся корпоративные версии ROSA Enterprise; наконец, на их основе создаются сертифицированные защищенные варианты — ROSA «Кобальт» (защита

О российском по «Быстро создать отечественные продукты „с нуля“ во всех областях — утопия.»

варду Сноудену, например, об американской программе PRISM. Из громких историй конкретной реализации — атаки на иранские атомные объекты с помощью StuxNet. Что касается более мягких способов — это прекращение поддержки зарубежного ПО для российских организаций, попавших в «санкционные списки», таких как Банк «Россия» и СМП-Банк.

LXF: А что можно сказать о разработке «с нуля» российского ПО?

ВР: В России есть ряд успешных производителей ПО, создавших отличные продукты «с нуля» — например, 1С, АBBYY, Kaspersky, InfoWatch и др. Однако быстро создать отечественные продукты «с нуля» во всех необходимых областях — утопия, даже при наличии «бесконечного» финансирования: никакие деньги не заставят женщину родить ребенка раньше минимального срока. В создании ПО «с нуля» — аналогичные ограничения. Однако здесь может прийти на помощь свободное ПО, которое можно использовать как готовый плацдарм



конфиденциальной информации) и РОСА «Хром» (работа с государственной тайной).

LXF: Учитывая международный аспект разработки открытого и свободного ПО, скажите, как видится продолжение развития отечественных разработок в случае форс-мажорных ситуаций на мировой арене? Как она гипотетически может повлиять на контроль над разрабатываемым и используемым ПО?

ВР: Как и в случае с применением ядерного оружия, надеюсь, форс-мажорной ситуации с ПО никогда не случится, и мы будем иметь возможность разрабатывать программы совместно с мировым сообществом — так лучше и дешевле. Однако иметь резервный вариант для гарантии цифрового суверенитета страны —

необходимо. Для этого достаточно обеспечить инфраструктурные объекты, органы государственного управления и обороны программным обеспечением, для минимальной поддержки, ис-

О РАЗРАБОТКЕ ПРОГРАММ

«Разрабатывать программы совместно с мировым сообществом — лучше.»

правления ошибок и развития которого достаточно ресурсов внутри страны. С организационно-технической точки зрения это возможно, когда для конкретного ПО внутри границ страны доступны три таких составляющих:

■ 100% исходного кода;

■ средства разработки и сборки этого кода;

■ специалисты, которые разбираются в пп. 1 и 2 и могут с их помощью исправлять и развивать продукт.

LXF: Если свободное ПО играет такую важную роль в плане обеспечения независимости и безопасности страны, то каким образом можно способствовать процессу его качественной разработки на государственном уровне?

ВР: Поддержка государства может выражаться, прежде всего, в обеспечении приоритета спроса на отечественное ПО (как закрытое, так и на базе свободного). Второстепенные меры могут включать частичное финансирование создания и развития открытых технологий, поддержку интеграции российских специалистов с мировыми сообществами разработки. **LXF**



Анатомия дата-центра

Нейл Мор наведлся в центр обработки данных UKFast, чтобы узнать, как там заставляют тикать петабиты.

Заставить работать Интернет — задача не из легких, и машины, которые двигают Глобальную Сеть, скрыты за колючей проволокой. Что, естественно, делает их еще интереснее. И поэтому, в жажде узнать, что же нужно для создания современного ISP, *Linux Format* рванул в солнечный Манчестер, чтобы установить местонахождение его серверов, глубоко скрытых внутри новейших хостингового комплекса при UKFast. Мы спросили: как вы создаете ISP? Ответ оказался чертовски сложным.

Все мы ежедневно имеем дело с Интернетом. Иногда вы об этом даже не подозреваете; но вводите ли вы адрес в браузер, отправляете ли сообщение со своего телефона или покупаете пакетик чипсов, вы выходите на определенный уровень взаимодействия с Интернетом. Когда он работает, это фантастика. Доступ к онлайн-хранению данных, электронной почте, обмену быстрыми

сообщениями... да к чему угодно мы принимаем как нечто само собой разумеющееся. Когда он не работает, выражения, которые слетают с наших уст, могут вогнать в краску даже нашего художественного редактора-мексиканца.

Но почему он не работает? Что ломается между браузерами и сервером?

Огромную часть того, что мы называем Интернетом, составляют центры обработки данных. Это места, где размещаются, работают и получают поддержку огромные количества серверов со всего мира. То, что работает на этих серверах, может варьироваться от чьего-то персонального *OwnCloud* до финансовой системы мультинационального банка, обрабатывающей миллиарды транзакций ежедневно.

Самому центру обработки данных это безразлично: он бесстрастно размещает серверы, подает на них питание, охлаждает, чтобы они эффектив-

но работали и были постоянно соединены с большим Интернетом, независимо от того, идет ли снег, или дождь, или происходит землетрясение, или всеобщий сбой электричества. Не все центры обработки данных построены одинаково, и мы покажем вам, почему это должно вас заботить. Итак, войдем же в один из новейших центров обработки данных Великобритании, управляемый UKFast, и посмотрим, что же заставляет его работать!

Сеть, облако, большинство онлайн-сервисов и большая часть самого Интернета работает на системах GNU/Linux и FOSS. Мы это знаем, и вы это знаете. Но на чем это все работает? Или, скорее, где это все работает? Большая часть ответа приходится на центры обработки данных. Огромные здания, внутри которых сотни — нет, тысячи! — серверов круглосуточно и неумолчно жужжат все 365 дней в году плюс еще лишний день в високосный год.



Вот это жужжание и есть самое важное, потому что Интернет совершенно не выносит простоя. Простой стоит людям денег и поднимает им давление до критического уровня. Ели вы не вполне поняли — простой означает, что вы видите сообщения: “404 page not found [Страница не найдена]” и “host unreachable [Хост недоступен]”. И это явно не то, чего вы ожидаете от Google, Ubuntu, Red Hat или IBM, от чьих бесперебойных сервисов зависят миллионы пользователей. Эта проблема начинается выходить на первый план в саморазмещенных сервисах — компаниях малого и среднего масштаба, где необходимость из экономии поступаться удобством обуславливает выбор менее надежных опций.

UKFast была основана в 1999 году его руководителем Лоуренсом Джонсом [Lawrence Jones] и коммерческим директором Гейл Джонс [Gail Jones], его женой. Начиналось все как компания интернет-хостинга, и ее главный офис всегда располагался в центре Манчестера. Сейчас UKFast размещается в районе Birley Fields (это часть быстро развивающегося городского технологического центра), удобно устроившись рядом со зданиями университета Манчестер Метрополитан [Manchester Metropolitan University].

UKFast появилась на рынке центров обработки данных только в 2011 году, но уже оставила свой след в мире центров обработки данных. Это была первая хостинг-компания в Великобритании, которая в 2010 году получила сертификат за 100% нейтральный показатель уровня эмиссии углерода по сертификации PAS2060. Она получила множество наград за хостинг и была «Лучшим хостинг-провайдером [Best Hosting Provider]» четыре года подряд. Нам также известно, что она пропагандирует GNU/Linux и FOSS, предоставляя обучение и тренинги LPIC всем сотрудникам своей компании, и предлагает дни разработчика своим администраторам Linux. Так что она выглядит идеальной компанией для выяснения того, как работает и что собой представляет хороший дата-центр.

Нам устроили экскурсию по центру, которую провел руководитель отдела продаж UKFast, Руди Фейцма [Rudi Feitsma], и мы смогли увидеть, как

организована вся работа. Все начинается в Кампусе UKFast площадью 50000 кв. футов. В основных офисах работает 200 человек, и здесь имеются обычные отделы маркетинга, продаж и управления плюс отдел кадров — как в любой крупной компании. Помимо всей этой обычной деловой инфраструктуры, компания обязана обеспечивать поддержку поставляемых ею продуктов 7/24 круглый год. Для этого нужна команда обученных, квалифицированных и имеющих все нужные ресурсы администраторов, которые постоянно в курсе самых последних достижений в разработке.

Серебряные серверы

Это первое отличие центров обработки данных. И с этим мы сталкивались много раз во время нашего визита — разница в ресурсах, которые используются в заслуживающих награды центрах обработки данных, и тем, что вы можете установить у себя дома, в офисе или на среднем предприятии. Вы хотите сами потратить время, деньги и знания на реализацию таких систем или доверите это экспертам? Для нас и нашего дома это нормально; а что будет, если на кону вся инфраструктура бизнеса?

«То, что мы видим здесь [в UKFast], можно ужать до домашнего дата-центра, — говорит Руди Фейцма, — ведь применяются те же самые элементы. Однако в некий момент становится ясно, что с финансовой точки зрения это невыгодно. Вы видите, как очень многие работники сферы ИТ идут по этому пути просто потому, что им это

«То, что мы видим здесь, можно ужать до домашнего дата-центра.»

хочется. Они желают создать центр обработки данных и работать с ним. Они идут к своему руководителю и говорят: „Мы должны это сделать, он должен у нас быть“. Они строят центр обработки данных, тратя на это большие деньги, и обнаруживают, что финансово это невыгодно для компании. С технической точки зрения это может быть очень интересно, и он даже может работать; но зачем же мучиться, когда уже есть множество дата-центров вроде нашего?»

С чего начинается создание дата-центра? С энергоснабжения. Оно используется для обеспечения работы всех серверов и системы охлаждения. Электроподстанция — единственный объект на территории UKFast, который не принадлежит дата-центру. С подстанции трансформатор берет энергию и подает ее на энергоблок стоимостью в миллионы фунтов. Именно здесь вы и получаете первое представление о том, насколько серьезным предприятием является дата-центр. Внутри находится панель с регуляторами мощности длиной почти с междугородний автобус, ряд источников бесперебойного питания (UPS) и панель свинцово-кислотных батарей на многочисленных полках.



Охлаждение серверов.

Регуляторы мощности обеспечивают ровную работу всех стоек. Если произойдет сбой электросети, они обеспечат переключение на резервные батареи, и три дизельных генератора, которые обычно спокойно сидят снаружи, оживут. Комната энергоснабжения — единственная, которая охлаждается полностью, и вы можете в прямом смысле ощущать силу ветра, дующего по ней — в основном центре обработки данных поток воздуха проходит под полом и непосредственно через стойки.

Весь комплекс построен по схеме N+1, и если одна установка ломается, остальные части без проблем берут на себя работу. Это объясняет, почему снаружи здания в ряд поставлено три зеленых ящика высотой в два этажа и размером с контейнер для морских перевозок. Каждый из них содержит промышленный дизельный генератор, специально разработанный для использования в дата-центре. Каждый из генераторов заправлен топливом по 50000 литров, и они готовы ожить и начать вырабатывать энергию через 56 секунд по отключении электричества. Топливо проверяется еженедельно, так же, как и машины, поэтому UKFast знает, что их решение работает и готово запуститься в случае необходимости.

Как отмечает Фейцма, в создании внутреннего дата-центра и экономии на UPS нет никакого смысла: «но именно это и происходит, поскольку UPS нынче дороги, сложны в эксплуатации, или их просто нельзя поместить в должное окружение, как эти; а такое может привести к тому, что кто-то просто пройдет мимо и выдернет кабель».

Внутри кожухов генераторов дюйм за дюймом проложена звукоизоляция. Сами генераторы круглосуточно и ежедневно стоят на дежурстве, чтобы при необходимости включиться немедленно, и их опробывают ежедневно, чтобы убедиться в их готовности к работе.

▶ Три массивных дизельных генератора позволяют центру работать в случае отключения энергии.



Структура центра обработки данных UKFast оптимизирует поток воздуха для охлаждения посредством размещения всех кабелей и проводов над серверами. Воздух идет под полом и через решетки. Он втягивается серверами спереди и попадает в нагретый проход за ними. Охлаждение происходит непосредственно в каждом отсеке, и ряды отсеков снаружи запечатаны. Вы можете ощущать, что они работают, потому что снаружи всех рядов жарко, но открыв дверь в конце и войдя внутрь, вы вмиг обнаружите, что там весьма прохладно и очень шумно.

Более старые дата-центры используют неэффективное охлаждение всего помещения или помещают кабели и провода под полом. Однако это нередко приводит к возникновению проблем, потому что по мере добавления новых кабелей становится все меньше места для них, пока решетки

в полу не выгибаются дугой и не становятся непригодными. Размещение кабелей вверху, под потолком, позволяет решить эту проблему и оставляет пространство под полом свободным для лучшего охлаждения.

Собственные серверы UKFast работают в одном здании и доступны только для персонала. Од-

«Если это не противозаконно и не пожароопасно, делайте что вам угодно.»

нако это также означает, что клиенты устанавливают и эксплуатируют серверы в другом здании, надежность и безопасность которого обеспечивается отдельно.

Наборы данных

«Это похоже на отель, — объясняет Фейцма. — Коль скоро это не противозаконно и не пожароопасно, делайте все, что угодно — нас это не касается. Наше дело — поддерживать охлаждение, энергообеспечение и соединение, а чем вы с ними занимаетесь — дело ваше. У клиентов даже замки свои собственные. Это то, что именуется независимыми от носителя дата-центрами: вы сами выбираете, кто предоставляет вам данные, а мы соединяем вас с ними несколькими способами. У нас уже есть здесь множество носителей, и мы протягиваем кабель от их коммутатора к вашему коммутатору. Если они не здесь, у нас есть очень умная система под названием DWDM (Dense Wavelength Division Multiplexing, мультиплексирование с уплотнением и разделением по длине волны), которая соединяет нас с другими центрами обработки данных. Это волоконная сеть. Затем мы можем выбрать носители в других центрах и принести их назад — именно так все дата-центры соединяются между собой».

Соединенный с клиентом хостинг — это то, что называется «теплый прием [meet and greet]» и «подсобка [build room]». Инженеры, работающие на собственном оборудовании, отказываются ходить с затычками, а уровень шума в серверной опасно высок, отсюда и необходимость в отдельной подсобке, чтобы не рвались барабанные перепонки и во избежание усталостной травматикки. Именно здесь хранится все оборудование принадлежащего сети носителя — эти стойки с коммутаторами, которые отдельно хранятся внутри закрытого решеткой

Анатомия центра обработки данных

1 Резервные свинцово-кислотные батареи
Хотя центр обработки данных UKFast защищен от долгосрочных отключений электроэнергии наличием дизельных генераторов, все же им нужны те самые эпические 60 секунд, чтобы включиться. И как же обеспечить защиту на этот промежуточный период? Целая стена свинцово-кислотных батарей предлагает достаточно энергии, чтобы центр продолжал работать в полную мощность в течение десяти минут — более чем достаточно, чтобы включились и заработали генераторы. И мы готовы поспорить, что пользователи этого даже не заметят!

2 Физическая безопасность
Дата-центр окружен забором, на который невозможно залезть, к тому же усиленным колючей проволокой, и центральные ворота открываются дистанционно. Потенциальным террористам стоит иметь в виду, что эти ворота могут без проблем выдержать лобовое столкновение с 10-тонным грузовиком, едущим со скоростью около 50 км/ч. И хотя центр обработки данных не является военным объектом, он располагается в анонимной индустриальной

зоне, так что вам придется провести целое расследование, чтоб хотя бы его обнаружить.

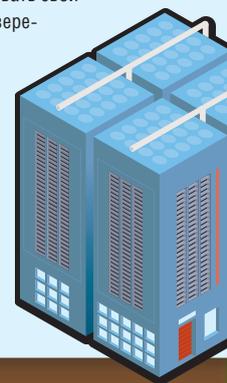
3 Кондиционирование воздуха
Большая часть энергопотребления центра обработки данных приходится на охлаждение, необходимое для того, чтобы не допустить перегрева всех систем. Оно обеспечивается большим количеством интегрированных систем кондиционирования воздуха. Вытяжные трубы расположены снаружи здания, и вытяжка работает только когда необходимо охлаждение. В прошлом коэффициент мощности центра был 2:1 — 2 кВт охлаждения на каждый 1 кВт, используемый для обработки данных. Однако для современной улучшенной конструкции он близок к 1:1.

4 Противопожарная система
Вокруг центра размещены большие красные цилиндры с трубками, подведенными к системе разбрызгивания. Это не водополивочные системы, а часть химической системы пожаротушения. Не волнуйтесь — вместо хладона система использует

другие химические вещества, которые позволяют человеку дышать. Это всяко лучше, чем заставить пожарных карабкаться по горам трупов.

5 Круглосуточный надзор
Помимо физических систем безопасности, весь центр круглосуточно охраняется и патрулируется. Внутри и снаружи здания размещены видекамеры, и все посетители регистрируются на входе и выходе. Чтобы получить доступ к своим серверам, вам необходимо зарегистрировать свой визит заранее и предъявить свое удостоверение личности с фото, и только тогда вас пропустят.

6 Банки серверов
Внутри всех этих систем охлаждения и безопасности размещается самое важное — серверы. Ряды серверных отсеков содержатся в закрытых «коридорах». Это помогает увеличить эффективность охлаждения, поскольку



отсека. «Итак, Cable & Wireless, Vodafone, Virgin и BT [мобильные операторы] могут иметь точку присутствия, — говорит Фейцма. — Им не нужно много оборудования, но они хотят, чтобы оно было отдельным. Захоти вы организовать соединение с Virgin, мы бы вывели соединительный кабель из их бокса через решетку в ваш бокс. И вся их сеть стала бы вашей. Если они начнут расширяться, они просто могут добавить еще один коммутатор, а потом еще один. Это на самом деле очень понятный способ, и делать такое им придется только однажды».

«Удобно также и то, — говорит Фейцма, — что территория хорошо подходит для центров обработки данных: здесь хороший волоконный доступ. Множество волоконных компаний уже здесь. Если бы мы, открывая свой центр обработки



➤ А если вас заботит регулирование, то ничто не может быть лучше этих регуляторов.

данных, выбрали место вроде Экклса [Eccles, пригород Манчестера, знаменит своими пирогами, но отнюдь не сетями], то по удобству соединения это было бы ничто. Пришлось бы потратить сотни тысяч фунтов на рытье траншеи под кабели, чтобы соединиться с кем-то поблизости. Если разместить свой центр в неправильном месте, единственный фактор сразу может перечеркнуть вашу перспективу стать серьезным игроком».

Поддерживая связь

Аспект удобства соединения — еще один важный элемент. В своих дата-центрах UKFast может использовать свое волокно, соединяя вас с любой требуемой вам сетью, даже если у нее нет точки присутствия в их комнате «теплого приема». Продолжая систему N+1, UKFast использует различные волоконные сети. Две 10-ГБ волоконных сети выходят из центра в двух разных направлениях через два коммутатора Cisco провайдерского уровня. Все серверы и клиентское оборудование имеют доступ к разным сетям, и каждый подключен к обоим коммутаторам. И если произойдет сбой у одного, другой обеспечит работу.

Фейцма говорит, что дело не просто в том, чтобы заявить «вот пустое здание, давайте сделаем здесь дата-центр»: «Вам нужно энергоснабжение и удобное соединение; об охлаждении говорят многие, но это также и просто поток воздуха. И вместе с безопасностью это составляет четыре аспекта центра обработки данных. За последние полтора года в Лондоне появилось множество дата-центров. Очень многие владельцы недвижимости решили, что могут неплохо на этом заработать, однако это будет сложнее, чем недвижимость. Крупные операторы в Докленде и Слау имеют мега-центры и взимают за это высоченную



➤ Эти генераторы на самом деле весьма серьезно... генерируют.

плату. У них есть международные банки, международные финансовые компании, игровые компании, компании игрового бизнеса».

Это впечатляющая организация во всех деталях, и, как говорит Фейцма, на нее не очень интересно смотреть, но вникнуть в нее очень трудно. Вы можете попытаться скопировать эту инфраструктуру в свой собственный бизнес, но все это вы получите — или чего вам следует ожидать — от центра обработки данных.

«Очень многие работники сферы ИТ никогда не были в дата-центре, — говорит Фейцма. — Лично я не бывал. Я 30 лет проработал в ИТ, но только в последние пять лет работаю с центрами обработки данных. Много лет назад я продавал технологии колл-центров, и им нужно было ПО для установки. А потом они сообщали, что оно не работает, и к ним надо было прийти, и их серверы стояли под лестницей и перегревались, или были соединены через модем 56К. Если вы собираетесь делать во что-то вложения, надо делать вложения правильно». LXF

кондиционирующий воздух доставляется в каждый отдельный отсек, а не во все помещение. Снаружи этих рядов воздух в серверной теплый, а внутри довольно холодно и ветрено.

7 Энергоподстанция

Единственный объект на территории, который не принадлежит UKFast. Именно отсюда электроэнергия подается на ее объекты.

8 Бесперебойное электроснабжение

Позади регуляторов мощности электролиний центра обработки данных ряд за рядом выстроились источники бесперебойного питания высотой в человеческий рост. Они установлены по принципу N+1, который означает, что все будет работать, даже если один из них откажет. Эти источники включаются в то самое мгновение, когда обнаруживается снижение напряжения в основной электросети, и в то же время они помогают пережить критический момент переключения между основной сетью, свинцово-кислотными батареями и дизельными генераторами.

9 Опорная сеть Интернет

Основная причина, по которой центр размещен в Манчестере — то, что в этом городе расположен опорный узел сети Интернет. Это сильно сокращает расходы по соединению дата-центра через волоконную сеть с высокоскоростной опорной сетью. Раньше дата-центры размещались исключительно в Лондоне, что приводило к солидным финансовым затратам, или в более дешевом месте за городом, что приводило к дополнительным расходам по обустройству предприятия оптоволоконной сетью.

10 Сетевые кабели

Чтобы данные могли курсировать между этими нудными серверами, нам по-прежнему нужны кабели. А точнее, многие мили кабелей Ethernet. Эти кабели проложены под полом множества дата-центров, однако это может привести к тому, что если их будет слишком много, то напольное покрытие вздуется. Поэтому UKFast предпочитают размещать кабели над серверами. Это выглядит очень опрятно и позволяет потоку воздуха проходить под сервером для более эффективного охлаждения.

11 Внутренняя безопасность

Чтобы пройти через внутренние и внешние двери, нужен пропуск. Это ограничивает доступ; все входящие и уходящие также фиксируются автоматической камерой.

12 Резервные дизельные генераторы

Что происходит, когда гаснет свет? Так вот, он не гаснет. В центре есть ряд резервных генераторов, которые автоматически включаются в то же самое мгновение, когда отключается электричество. Цель этих генераторов — обеспечивать энергию вместо основной сети электроснабжения в течение драгоценных 60 секунд. В UKFast не менее трех дизельных генераторов в ждущем режиме 24/7, 52 недели... ну, вы поняли суть. Эти генераторы еженедельно тестируются, и у них всех наготове 50 000 литров дизельного топлива. Звучит довольно просто, но иначе не выходит. Если у вас будет меньше генераторов, или вы не будете их проверять или обеспечивать топливом, то в тот самый момент, когда они вам понадобятся, они просто с треском взорвутся. Не лучшая ситуация для дата-центра.



СТАВИМ Linux на НОВЫЙ Chromebook



Тех, кто купил Chromebook и нуждается в полноценной ОС, **Нейл Ботвик** научит, как установить и пользоваться целым ассортиментом дистрибутивов Linux.

Chrome OS блистательна — для тех пользователей, на которых в общем и рассчитан Chromebook: она делает именно то, что и должна делать. Быстрая и простая в использовании — чего еще желать? Однако через некоторое время вы, возможно, поймете, что вам не хватает ряда функций, характерных для более традиционных операционных систем. Но не переживайте, потому что помощь рядом, в виде *Crouton*.

Crouton — это набор программ, которые настраивают среду chroot внутри Chrome OS, откуда вы можете запустить Linux OS — сейчас есть поддержка Debian и Ubuntu. Chroot — это не то же самое, что виртуальная машина: вы по-прежнему работаете в стандартной ОС, но в новой среде. У этого метода есть ряд ключевых преимуществ: например, не затрагивается существующая ОС, что облегчает возможность возврата к ней; используются драйверы Chrome OS для видео, беспроводных и прочих устройств, так что у вас не будет проблем с совместимостью; и он создан авторами Chrome OS, так что останется совместимым с будущими

обновлениями. Единственным настоящим недостатком использования *Crouton* является то, что вы столкнетесь с небольшим падением производительности; но вы ведь в любом случае покупали Chromebook не ради его умопомрачительной скорости. На случай, если вам интересно, *Crouton* — это сложный акроним (означающий ChRomium Os Universal chroot EnvirONment), который явно притянули за уши уже после того, как он появился.

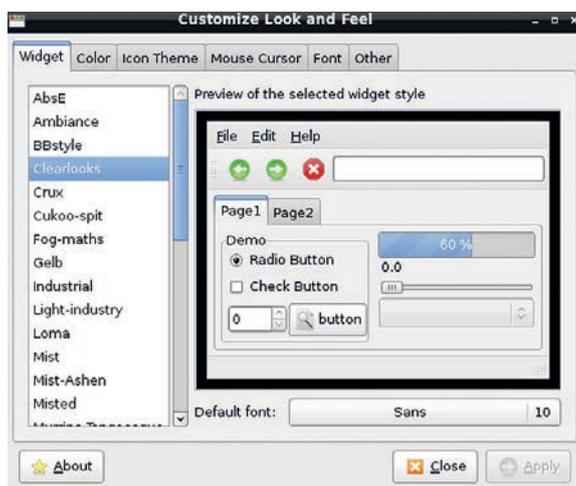
Прежде чем устанавливать другие дистрибутивы, неплохо было бы создать спасательный диск для восстановления вашего Chromebook, если что-то пойдет наперекосяк. Да и без всяких переустановок это все равно разумно сделать, особенно учитывая, насколько это просто: все, что вам нужно — USB-брелок или SD-карта не менее 2 Гб. Благодаря облачной природе Chrome OS, 2 Гб будет достаточно, потому что вам нужно только создать резервную копию операционной системы — ваши данные и настройки находятся в безопасности на серверах Google. (Подробности см. в пошаговом руководстве *Диски восстановления* на стр. 53).

Изучим жаргон!

apt-get

Программа, используемая для установки программных пакетов в Debian, Ubuntu и прочих дистрибутивах Linux.

➤ На Chromebook работает LXDE, но Chrome OS все еще там.



Вдарим по оболочке

Когда будете готовы, начните скачивать *Crouton* с <http://goo.gl/fd3zc>. Это скрипт, который добудет и установит все, что вам нужно. Вы запускаете его из оболочки — да, оболочка на Chromebook есть! Нажмите на Ctrl+Alt+T, чтобы открыть оболочку *Crosh* во вкладке браузера. Оболочка весьма ограниченная, но запустите ее, чтобы вызвать нормальную оболочку *Bash*. *Crouton* должен знать, какой дистрибутив вы хотите установить; он вызывает эти релизы и выбирает их по параметру `-r`. Еще ему нужно знать целевую среду, которую вы хотите установить. Цель — это подборка программных пакетов, например, определенный рабочий стол. Следующие две команды выведут список опций:

```
sh -e ~/Downloads/crouton -r list
sh -e ~/Downloads/crouton -t list 2>&1 | more
```

Вторую команду следует перенаправить в файл `more`, поскольку она занимает несколько полных экранов — нажмите на пробел, чтобы прокрутить их все. Решив, какие вам нужны релиз и цель,

Включаем режим разработчика

Использование *Crouton* означает перевод вашего Chromebook в режим разработчика — Developer Mode, в котором вы получаете доступ root и даже оболочку *Bash*. И это не заплатка, а полностью поддерживаемая, хоть и скрытая опция. Предупреждение перед тем, как вы начнете: включение Developer Mode сотрет все, что у вас хранится. Ваше облачное хранение затронуту не будет, но все файлы, хранящиеся локально, должны быть переброшены на Google Drive до того, как вы начнете работу. Способ включения режима разработки зависит от устройства — вы найдете инструкции на сайте *Chromium* здесь: [http://](http://bit.ly/1gDHPGd)

bit.ly/1gDHPGd. На Acer C720, который мы использовали для тестирования, как на большинстве устройств Samsung, вы выключаете устройство и затем удерживаете клавиши Esc и Refresh перед тем, как нажать на кнопку питания. Возникнет экран восстановления; затем нажмите на Ctrl+D, чтобы включить Developer Mode. На других устройствах для этого предусмотрена специальная кнопка.

Включив Developer Mode, вы увидите, что подтверждением ОС при каждом включении будет экран OFF; нажмите Ctrl+D для продолжения загрузки или подождите 30 секунд.



➤ Загляните на www.chromium.org, чтобы выбрать включатель Developer Mode для своего устройства.

запускайте *Crouton*. Чтобы, например, установить Ubuntu 13.10 (Saucy Salamander) с рабочим столом Unity, скомандуйте

```
sudo sh -e ~/Downloads/crouton -f saucy -t unity
```

Здесь приставлено **sudo**, так как программы устанавливаются от имени root. Целей можно указать и несколько, как в следующем примере, который устанавливает Debian Wheezy с рабочим столом *LXDE* и медиа-центром *XBMBC*:

```
sudo sh -e ~/Downloads/crouton -f wheezy -t lxde,xmbc
```

Начало

Время установки зависит от выбранной цели и скорости вашего интернет-соединения, и установка может затянуться. По ее окончании вам сообщат, какие требуются команды, чтобы запустить выбранный дистрибутив в *chroot* — например,

```
sudo startunity
```

Эта команда приведет вас на стандартный рабочий стол Ubuntu. Закончив, выйдите из системы стандартным образом, и вы вернетесь в привычную Chrome OS. Вы можете переключаться между ними, удерживая Ctrl+Alt+Shift и нажав одновременно Forward или Back. Вообще-то кнопки навигации над цифровым рядом Chrome OS считаются кнопками F в Linux, поэтому на самом деле это Ctrl+Alt+Shift+F1 и Ctrl+Alt+Shift+F2.

Установка, которую вы получаете — это не тот полный дистрибутив, как если бы вы установили его изначально, но все дополнительные пакеты можно добавить обычным способом. При использовании Unity не устанавливается Software Centre, поэтому откройте терминал в Unity (Ctrl+Alt+T) и запустите

```
sudo apt-get update
```

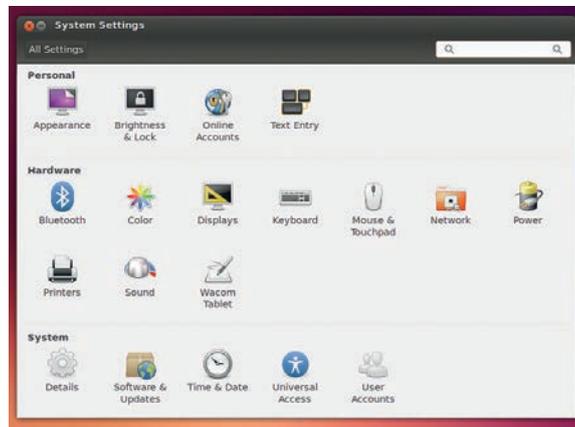
```
sudo apt-get install software-center
```

Теперь вы можете установить любые другие нужные вам пакеты из GUI. Вы можете также установить дополнительные целевые среды с помощью флага **-u**. Например, чтобы добавить среду *LXDE* в Ubuntu *chroot*, созданную ранее, следует запустить

```
sudo sh -e ~/Downloads/crouton -f saucy -u -t lxde
```

Добавим конфиденциальности

Как вы, вероятно, заметили, включение Developer Mode даст вам доступ root через **sudo**, не требуя пароля. Это чуть менее безопасно для Chrome OS, но ваш логин и файлы по-прежнему защищены вашим логином Google, хотя это означает, что все файлы в вашей *chroot* читаются даже с гостевого логина без пароля. Если вас это беспокоит, можете зашифровать всю *chroot* с помощью флага **-e** для *Crouton*. С вас требуют пароль, который будет использован для шифрования всей директории *chroot*, и это означает, что вы не сможете ни читать, ни запускать *chroot* без указания пароля. Например:



➤ Unity идеально подходит для работы в полноэкранном режиме.

Изучим жаргон!

chroot
Директория, в которой программа заперта так, что не видит ничего за ее пределами.

```
sudo sh -e ~/Downloads/crouton -e -r wheezy -t xfce
```

Есть множество релизов дистрибутивов и целевых сред на выбор. Вы можете установить их все сразу, но это будет довольно громоздко. Так как же вам все их испробовать? Решение — завести столько *chroot*'ов, на сколько у вас хватит места.

Если вы планируете это сделать, вам, возможно, будет проще использовать опцию *Crouton -n*, чтобы дать каждому *chroot* имя; иначе они просто получат имена релизов. Названия очень важны при установке нескольких релизов, поскольку они потребуются для команд запуска, не то *Crouton* просто загрузит первый же релиз, в котором найдет заданную вами целевую среду.

Добавление **-n**, подобным образом, позволит вам обеспечить загрузку нужного релиза:

```
sudo startunity -n saucy
```

Crouton также установит пару полезных инструментов — в частности, *edit-chroot*. Его можно использовать для резервного копирования *chroot*.

```
sudo edit-chroot -b saucy
```

создаст резервную копию в **~/Downloads**, которую можно восстановить с помощью

```
sudo edit-chroot -r ~/Downloads/backup-file.tar.gz
```

Скопируйте в безопасное место. Даже если вы сделаете полный перезапуск/восстановление системы, вы все равно можете вернуть *chroot*, снова скачав *Crouton* и запустив

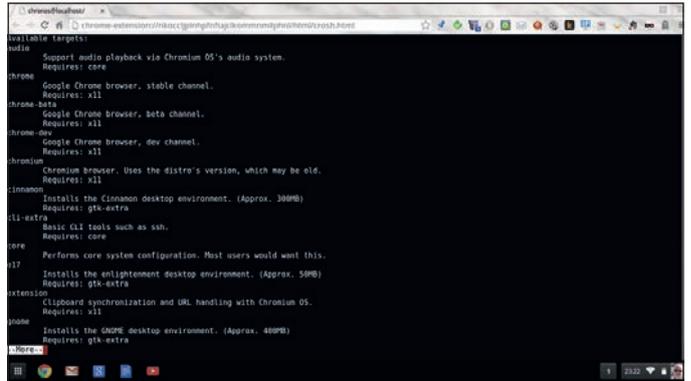
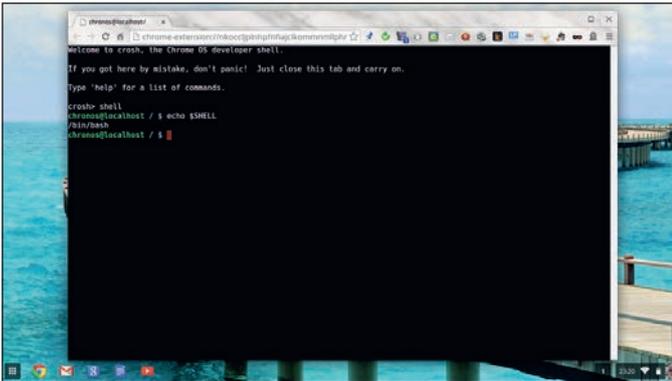
```
sudo sh -e ~/Downloads/crouton -f backup-file.tar.gz
```

Вы также можете использовать *delete-chroot*, чтобы удалить *chroot* (о чем вы, вероятно уже догадались), или можете просто удалить директорию, содержащую его, из **/usr/local/chroots**, чтобы вернуться в основную Chrome OS. Естественно, если вы этого хотите. Следуйте инструкциям на следующей странице... ➤

Скорая помощь

Пробуя несколько дистрибутивов или целевых сред, удаляйте те, с которыми работа уже закончена. Иначе, поскольку каждый из них занимает несколько ГБ, у вас скоро закончится диск.

Установка релиза

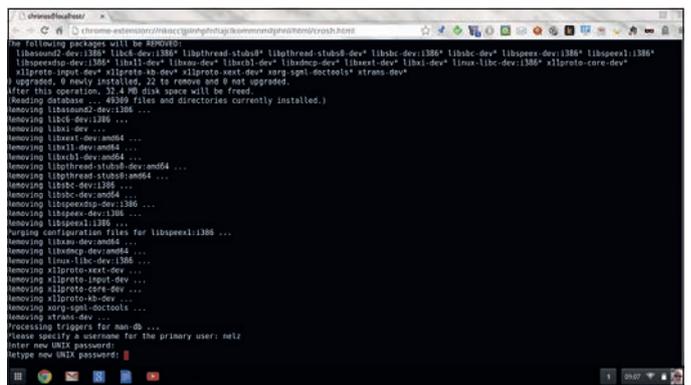
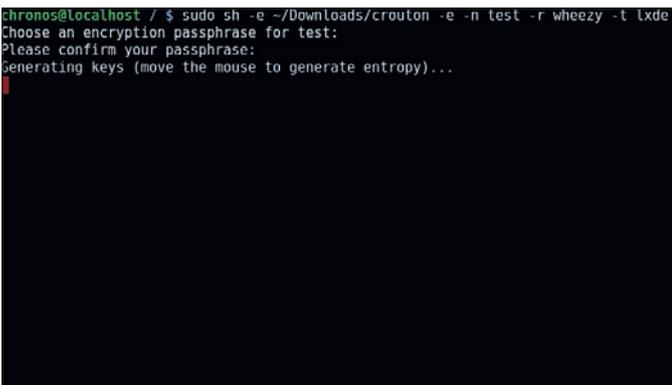


1 Откройте оболочку

Откройте окно терминала, нажав на **Ctrl+Alt+T**. Это будет стандартная оболочка *Crosh* во вкладке браузера с ограниченным набором команд — вы можете просмотреть их, введя `list`. Одна из команд — `shell`, она дает вам полную оболочку *Bash*, как в других дистрибутивах. Это правда — у Chrome OS за сценой скрывается настоящая Linux OS.

2 Выберите релиз и целевую среду

Запуск *Crooton* с параметрами `-t list` покажет вам все имеющиеся целевые среды. Возможно, вам понадобится одна из стандартных сред рабочего стола. Chrome-возок не отличается большой мощностью, поэтому разумным выбором будет легкий рабочий стол типа *LXDE*; а *Unity* лучше подходит для работы в полноэкранный режиме.

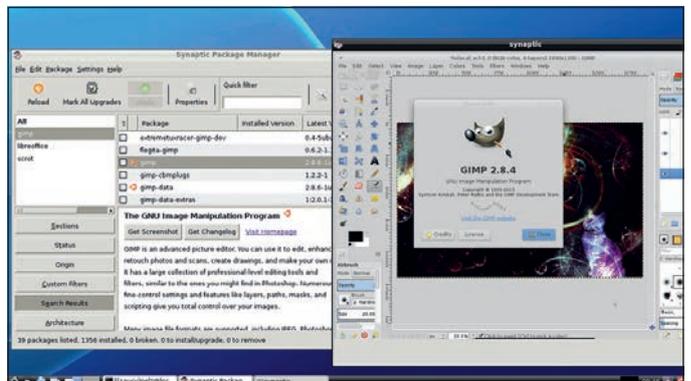
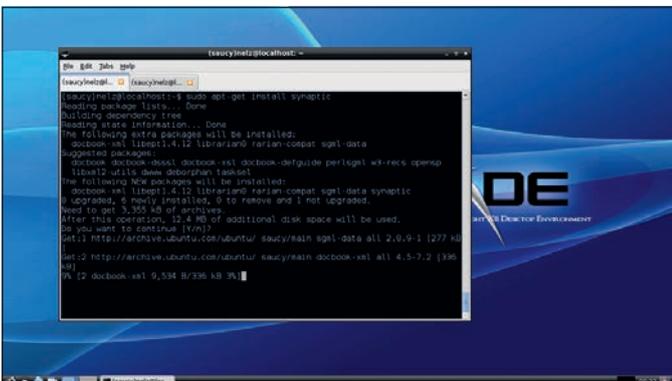


3 Шифруйте свои файлы

Добавление `-e` в командной строке *Crooton* (это не то же самое, что `-e`, следующее за `sh`) приведет к тому, что ваш `chroot` будет храниться в зашифрованной директории. Выберите достойный пароль — это единственная защита ваших файлов; но помните, что большая часть ваших данных будет, вероятно, храниться в облаке, потому что у Chromebook очень ограниченные возможности хранения.

4 Установка дистрибутива

Поскольку *Crooton* — это только программа установки, ей нужно предварительно скачать файлы релиза дистрибутива, так что дайте ей на это время. Даже с быстрым соединением на это может уйти более 30 минут, чтобы скачать и установить все, если вы выбрали большую целевую среду — размеры указаны в итоговом списке по `crooton -t list`.



5 Добавьте менеджер пакетов

Целевые среды по умолчанию включают только менеджер пакетов командной строки, `apt-get`. Для большинства первым шагом будет открыть терминал и использовать его для установки более дружелюбной опции, например, *software-center* для Ubuntu или *Synaptic* для Ubuntu или Debian. Запустите `sudo apt-get update`, чтобы обеспечить себе свежую версию, а затем `sudo apt-get synaptic`.

6 Запуск Synaptic

Установив *Synaptic*, вы получите легкий доступ ко всем программам в репозитории дистрибутива. Большинство целевых сред урезаны, чтобы сэкономить на скачивании и получить более быструю установку, но вы можете установить отсюда все что угодно. Используйте кнопку **Search** или просто просмотрите категории, чтобы увидеть, что здесь есть.

ДИСКИ ВОССТАНОВЛЕНИЯ

Create OS Recovery Media

If you ever need to restore your computer's operating system, you'll need a memory stick.
[Learn more about system recovery](#)

USB memory stick detected

⚠ All files on U3_Cruzer_Micro will be erased.

Create OS Recovery Media

If you ever need to restore your computer's operating system, you'll need a memory stick.
[Learn more about system recovery](#)

Copying recovery image...

180 MB of 1.4 GB copied

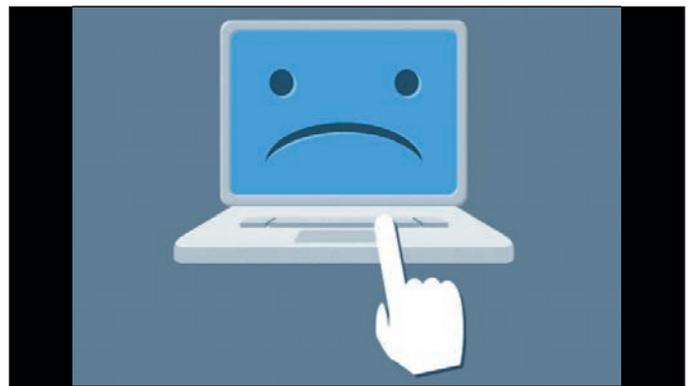
This may take a few minutes

1 Резервные копии на USB

Подключите USB-брелок или SD-карту объемом не менее 2 ГБ, откройте Chrome и введите `chrome://imageburner` в адресную строку. Chrome OS скачает и установит образ восстановления для вашего Chromebook. Если у вас более одной модели Chromebook, запустите это отдельно для каждой; вы получите правильные образы для всех устройств.

2 Создайте диск восстановления

После скачивания образ записывается на ваш USB-брелок. Если вы не создаете диск восстановления, можно также взять этот образ с другого компьютера и скопировать его вручную, следуя инструкциям на <http://google.com/chromeos/recovery>, но вы должны быть уверены в том, что у вас правильный образ — для разных моделей они отличаются.



3 В экстренном случае

Если вы повредите Chrome OS, и увидите жуткое сообщение: «Chrome OS отсутствует или повреждена [Chrome OS is missing or damaged]», подключите свое средство экстренного восстановления. При желании вы также можете выполнить принудительное восстановление, нажав на кнопку перезагрузки или на соответствующее сочетание клавиш, которое варьируется в разных моделях. Загляните в документацию вашего Chromebook, чтобы знать, что применять.

Crouton: За и против

Сравнение Chromebook стоимостью £200 с полноценным ноутбуком может показаться несправедливым — он скорее находится в ценовом диапазоне нетбуков и нацелен на ту же аудиторию; но мы все же попытаемся.

Запуск Ubuntu или Debian на Chromebook — это то же самое, что их запуск на обычном ноутбуке. Вся разница — в использовании chroot. То есть надо бы сначала загрузиться в Chrome OS, а затем открыть оболочку, чтобы запустить сессию в chroot; но Chromebook создан так, что его лучше перевести в спящий режим, чем выключать, и в этом часто нет нужды. Поскольку оборудование используется через Chrome OS, вам следует, например, настроить ваше сетевое соединение именно там,

но раз уж вы можете переключаться между своими ОС, это не проблема. Это не двойная загрузка, а работа с двумя ОС одновременно — намного удобнее.

Основное ограничение при таком раскладе — отсутствие места для хранения и зависимость от сетевого соединения и облачных сервисов. Хотя Chrome OS работает с ними весьма прозрачно, вам нужно настроить какую-либо онлайн-синхронизацию из вашего дистрибутива в chroot, используя сервисы вроде OwnCloud, Spidegoak или Dropbox.

Есть и другие способы установки Linux на Chromebook, но Crouton делает это в наименее навязчивой манере, не затрагивая вашу существующую ОС (разве что придется активировать Developer Mode). Возможность попробовать разные опции и удалить

их по окончании — тоже важное преимущество этого подхода. Опять же, Crouton — не сторонний продукт, а разработка авторов Chrome OS.

Будущие обновления Chrome OS могли бы временно повредить ваш chroot, но обновления через Crouton исправит это. **LXF**





Что за штука... **UEFI**

Александр Толстой рассказывает о современном способе загрузки компьютера — быстро и без привычных сообщений BIOS.

В Как всегда, начнём с вопроса «И что же это за штука такая?»

О Если коротко, то UEFI — Extensible Firmware Interface — интерфейс между операционной системой и микропрограммами, которые отвечают за низкоуровневые функции оборудования.

В Тоже как всегда, легче почему-то не сделалось...

О Ну, смотрите. Когда вы включаете компьютер или ноутбук, перед загрузкой ОС вы видите

некие сервисные сообщения — так даёт о себе знать BIOS материнской платы, микропрограмма, которая нужна для начальной загрузки компьютера, до того как управление передаётся ОС.

В А зачем операционной системе нужен какой-то BIOS? Без него никак?

О До недавнего времени было действительно никак. Видите ли, ОС должна как-то узнать о том, какое оборудование у вас установлено; это касается как компонентов, входящих в состав

компьютера, так и подключённых устройств. BIOS как раз предоставляет эти сведения для ОС, оставаясь при этом независимым от неё: BIOS записан в отдельный чип памяти на материнской плате. Иногда его ещё называют «прошивкой».

В Погодите. Вы сказали «до недавнего времени». Разве сейчас что-то поменялось?

О Именно так. Ещё в середине 1990-х годов компания Intel столкнулась с тем, что BIOS, изначально рассчитанный для машин IBM PC/AT,

не годился для серверных платформ на базе Intel Itanium.

В И что же в данном случае не устраивало Intel?

О В двух словах — малый объём адресуемой памяти (1 МБ), ограничение на 16-разрядный исполняемый код и другие аппаратные ограничения. В связи с этим, уже в конце 2000 года Intel выпустила первую спецификацию EFI, а в 2005 году передала развитие технологии в руки UEFI Forum. С этих пор и начинается история UEFI (Unified Extensible Firmware Interface — унифицированный и расширяемый микропрограммный интерфейс). Последняя версия спецификации (2.4) была утверждена в июле 2013 года.

В Давайте перейдём от скучной теории к сути. Кстати, в чём она?

О В первую очередь, EFI полностью раскрывает возможности 64-разрядных ОС, позволяя им загружаться и работать быстрее. Второй немаловажный момент связан с адресацией памяти. С EFI компьютер может поддерживать больше ОЗУ и жёсткие диски большего объёма. Теоретически максимальный размер жесткого диска может достигать 8192 эксабайта, что составляет примерно 8,8 триллионов терабайт, а это даже при нынешних объёмах передачи информации является весьма впечатляющей цифрой, особенно если учесть, что размер архива всего Интернета составляет 10 петабайт.

В Замечательно, но как это касается меня и моего ноутбука?

О Не торопитесь, это ещё не всё. Системы с EFI загружаются намного быстрее, чем компьютеры с традиционным BIOS. Это объясняется тем, что инициализация отдельных аппаратных компонентов производится параллельно. Кроме того, такие функции, как поддержка стека TCP/IP и простейшие файловые операции, реализованы в самой EFI и работают ещё до загрузки ОС. Более того, функциональность EFI может быть расширена за счёт сторонних модулей.

В Можно конкретный вопрос? Где это всё хранится?

О Базовая часть EFI записана в ПЗУ на материнской плате вашего ПК или ноутбука, а загрузчик ОС и дополнительные расширения хранятся на специальном EFI-разделе вашего жёсткого диска. Это должен быть небольшой раздел размером 200–300 МБ с файловой системой FAT32, находящийся в начале диска и имеющий специальную EFI-метку. Есть и ещё один важный момент. Помните, ранее я упоминал про расширенную адресацию жесткого диска? Для пользователей это

означает, что жёсткий диск должен быть размечен с использованием схемы GPT (GUID Partition Table), а не MBR (Master Boot Record).

В Я вряд ли буду сам размечать диск. Как мне понять, используется ли у меня EFI или нет?

О Если вы покупали свой компьютер не ранее 2011 года, то скорее всего поддержка EFI у вас имеется. Другой верный признак — наличие предустановленной Windows 8. Одно из требований Microsoft к производителям материнских плат состоит в поддержке Secure Boot — это специальный режим «безопасной» загрузки с использованием подписанного закрытым ключом загрузчика.

«UEFI актуально при сборке нового ПК или установке Linux на чистый ноутбук.»

Везде, где установлена Windows 8, используется Secure Boot, который является частью спецификации UEFI. Таким образом, на новых компьютерах EFI гарантированно есть.

В Microsoft? Чую, тут дело не только в безопасности...

О Вы скорее всего правы, т.к. Secure Boot затрудняет установку на ПК с Windows 8 альтернативных ОС, и только слепой не увидит здесь заградительный барьер, который нам подсовывают из Редмонда. К счастью, режим Secure Boot в большинстве новых материнских плат всё ещё может быть отключён, то есть вас вряд ли заставят пользоваться им принудительно. Но даже в этом случае не всё так плохо: для Linux имеется рабочий способ установки рядом с Windows 8, использующей Secure Boot.

В То есть я смогу поставить Linux второй системой рядом с Windows 8?

О Да, сможете. Для многих популярных дистрибутивов вы найдёте в Сети пошаговые инструкции. Однако если у вас установлена Windows ранних версий или же вы вообще хотите использовать только Linux, всё будет проще. В сервисном меню материнской платы (которую в народе и называют «биос») убедитесь, что опция UEFI включена, а Secure Boot — выключена, после чего приступайте к установке вашего дистрибутива. Но всё же, будьте осмотрительны...

В Что, опять?.. Как всегда, проблемы какие-нибудь?

О Поддержка UEFI реализована в Linux в целом неплохо, но различается в зависимости

от дистрибутива. Хорошо работает установка Ubuntu и его клонов, openSUSE, Manjaro, заявлена поддержка в Fedora и Росе. Лучше будет заранее уточнить в документации вашей системы, имеется ли поддержка UEFI и насколько она отличается стабильностью.

В И давно ли творится такая катавасия с этим UEFI?

О Вообще-то первый загрузчик с поддержкой EFI существует аж с 2000 года и называется *elilo*. Работа над доведением до ума *Grub* ведётся с 2008 года, и сейчас уже существует стабильный вариант *Grub2-EFI*. Вам следует знать, что UEFI

хоть и существует уже много лет, но на потребительском рынке устройства с поддержкой этой технологии появились относительно недавно (~3 года назад). Наконец, функциональность ПК с EFI ничем не отличается от функциональности ПК с традиционным BIOS, если отметить механизм начальной загрузки и средства диагностики, встраиваемые в оболочку EFI.

В То есть спешить с переходом на EFI не следует?

О Если у вас всё уже настроено и работает — однозначно менять ничего не стоит. Использование EFI актуально при сборке нового ПК или установке Linux на чистый новый ноутбук. Также, сообщество в своё время всколыхнуло жёсткое требование к Secure Boot со стороны Windows 8. С EFI вы получите очень быструю загрузку ОС, всё остальное же останется прежним.

В Последний вопрос: меня не покидают мысли о кознях Microsoft. Что скажете?

О Компания Microsoft — один из самых ранних и активных членов UEFI Forum, и она повлияла на его разработку очень сильно. У многих наблюдателей сложилось обоснованное мнение, что основная цель EFI — усложнить модификацию микропрограммы ПК для сторонних вендоров. Но особенно не волнуйтесь: последние релизы Windows 8.x чрезвычайно слабо показали себя на рынке, в то время как популярность настольного Linux продолжает неуклонно расти. Это поддерживает здоровую конкурентную среду, а значит, вы всегда сможете установить Linux на свой новый компьютер — будь то в режиме EFI или без него. **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.

Криптовойны

За несколько дней до того, как я начал писать эти строки, вышел OpenSSH 6.6. Одно из главных нововведений — несомненно, поддержка протокола обмена ключами Curve25519 Даниэля Бернштейна [Daniel Bernstein] с эллиптической кривой Диффи–Хеллмана [Diffie–Hellman]. То есть как — вы не поняли, о чем я? Это важная технология — извольте вникнуть!

Проблема в том, что 99,99% из нас не понимает, что происходит. Если армия А выходит на поле боя с копьями, а армия В — с гаубицами, нам ясно, что армия В скорее всего победит. Но что будет, если армия А применяет протокол Диффи–Хеллмана с эллиптической кривой для обмена симметричными ключами сеанса с тройным DES, а армия В использует алгоритм Ривеста–Шамира–Адлемана [Rivest–Shamir–Adleman] для обмена 256-битными симметричными ключами блочного шифрования Twofish?

Криптовойны с криптографами и криптоаналитиками на цифровом поле боя, где нет места человеку — это, несомненно, реальность. По ряду оценок, взлом немецкого шифра «Энигма» в Польше и позднее в Блетчли-парк в Великобритании сократил II Мировую войну на 2 года. США долго запрещали экспорт криптографических технологий, опасаясь, что попав не в те руки, они создадут военное преимущество, и относили их к «военному снаряжению». Но благодаря открытым программам типа PHP и OpenSSL передовые криптографические технологии перешли из рук военных в общее достояние.

Считается, что в Curve25519, вероятно, нет «лазеек» — намеренно внесенных уязвимостей, позволяющих агентствам вроде АНБ и Центра правительственной связи Великобритании перехватывать наши (якобы) приватные разговоры; но по правде говоря, я не слишком беспокоюсь о том, что кто-то в Центре правительственной связи читает мою переписку (ребята, привет!); кто бы это ни был, он скоро помрет со скуки. chris.linuxformat@gmail.com.

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

nftables

Сначала был *ipfwadm*, потом были *ipchains*, потом *iptables*, а теперь есть *nftables*.

Многим из вас неизвестна программа *iptables*, используемая для загрузки правил фильтрации пакетов в подсистему сетевой фильтрации ядра [netfilter] для настройки брандмауэра. Все изменится с появлением *nftables* — готовящегося к выходу проекта, который начался еще в 2009 году.

Кривая обучения при переходе с *iptables* на *nftables* оказывается довольно крутой, так как их синтаксис существенно различается.

Попробую дать вам о нем представление. Определим таблицу с одной цепочкой правил Input, привязанной к пути input подсистемы сетевой фильтрации ядра. Мы пропускаем на адрес обратной петли весь трафик и пакеты для нашего сервера SSH, и отбрасываем, подсчитываем и записываем в журнал все остальное:

```
table Filter {
  chain Input {
    type filter hook input priority 0;
    ct state established accept
    ct state related accept
    iif lo accept
    tcp dport ssh counter accept
    counter log drop
  }
}
```

Как видите, язык здесь значительно богаче по сравнению с *iptables*. Например, следующее правило заменяет три отдельных команды *iptables*:

```
nft add rule ip filter input tcp dport {ssh, http, https} accept
```

Список в фигурных скобках называется множеством. Можно определить собственные множества (например, черный список IP-адресов), давать им имена и использовать эти именованные множества в других правилах. При этом в черный список IP-адресов всегда можно добавить новый адрес.

Попробуйте сами

Если вы хотите попробовать *nftables*, то в настоящее время ваш выбор ограничен. Эта технология не входила в основную линейку ядра до версии 3.13. В Ubuntu 14.04 (на момент написания статьи релиз был все еще в бета-версии) используется ядро 3.13, но пользовательские библиотеки и утилиты для *nftables* еще не попали в стандартные репозитории. Я предпринял формальную (и неудачную) попытку загрузить исходники *nftables* и сопутствующие библиотеки из Git-репозитория и собрать их, но в конце концов нашел то, что мне было нужно, уже в PPA (персональном архиве пакетов) от Сю Чжэня [Xu Zhen] и (в свежееустановленной Ubuntu 14.04) смог установить и настроить *nftables*, употребив всего лишь три команды:

```
sudo add-apt-repository ppa:xuzhen666/nftables
sudo apt-get update
sudo apt-get install nftables
```

Где узнать больше

Документация пока что бедновата, но есть довольно подробная map-страница для *nftables*. Кроме того, на Regit.org (<http://bit.ly/nftablesshowto>) можно найти полезное руководство, а на <http://bit.ly/lovenftables> — дополнительное обсуждение с примерами. Домашняя страница проекта — <http://netfilter.org/projects/nftables>.



► В *nftables* используется существующая инфраструктура netfilter — существующие компоненты привязки, отслеживания соединений, пользовательских очередей и журналирования.

API и протоколы

Понятия API и «протокол» дошли до точки, в которой стали почти синонимами. Далее — устраняем путаницу...

Преподаватель не должен спорить со студентами. Это непрофессионально и снижает вероятность успешного окончания учебного курса. Избегание спора иногда требует такта (моя не самая сильная сторона), и на одном из недавних занятий я оказался вовлечен в... назовем это жаркой дискуссией о том, чем API отличается от протокола. Студент был уверен, что это синонимы, а я считал, что это совершенно разные вещи.

Вы уселись удобно? Тогда объясню. Мой верный Оксфордский словарь английского языка содержит не менее четырех определенных слова «протокол». Четвертое и самое подходящее нам гласит: «Набор правил, определяющих обмен или передачу данных между устройствами». Мир протоколов наполнен аббревиатурами. POP, PPP, HTTP, SMTP, TCP, UDP — список бесконечен (ну, вообще-то конечен, но уж точно очень, очень длинный). Каждый из этих протоколов — подробное формализованное описание «проводных» форматов сообщений, которыми обмениваются для выполнения определенной операции — отправки электронного письма, загрузки web-страницы и т.д.

Слово «простой [simple]» в названиях протоколов — распространенная и иногда раздражающая тенденция: так, у нас есть SMTP (Simple Mail Transmission Protocol — простой протокол передачи почты), SNMP (Simple Network Management Protocol — простой протокол управления сетью), SOAP (Simple Object Access Protocol — простой протокол доступа к объектам), SASL (Simple Authentication and Security Layer — простой слой аутентификации и безопасности) и многие другие. Большинство из этих протоколов далеко не просты, и непонятно, должно ли слово «простой» в их названии убеждать вас в том, что вы их поймете, или вызывать чувство неполноценности, если не поймете. Меня особенно позабавил ESMTP (Extended Simple Mail Transmission Protocol — расширенный простой протокол передачи почты). Неужели нельзя было назвать его хотя бы CSMTP (Complicated Mail Transmission Protocol — сложный протокол передачи почты) или остановиться на MTP (Mail Transmission Protocol — протокол передачи почты)?

Протоколы, применяемые в Интернете, описываются в стандартных документах, называемых RFC (request for comments — запрос комментариев). В них для описания формата пакетов традиционно используется «ASCII-стиль». В качестве несложного примера взгляните на RFC 1350, который определяет Тривиальный протокол передачи файлов (Trivial File Transfer protocol). Определения протоколов обычно довольно формальны и утомительно подробны. Идея состоит в том, что два человека могут взять по копии протокола и разойтись и вернуться с реализациями протокола, которые успешно взаимодействуют друг с другом — хотя не знаю, часто ли это обходится без доработки («Ах вот как ты это понял...»). Протоколам также обычно нет дела до языка, на котором они реализованы, и до операционной системы, на которой может быть запущена их реализация.

Протоколы — как луковицы

Протоколы обычно используются в сочетании, накладываясь один на другой послойно. Например, сообщения, определенные протоколом HTTP, передаются по TCP (Transmission Control Protocol — протокол управления передачей), а сами сообщения TCP, в свою очередь, передаются по IP (Internet Protocol — протокол Интернета). На стороне отправителя, когда пакет передается

на слой уровня ниже, к пакету добавляется заголовок, позволяющий этому слою нижнего уровня сделать свою работу. Таким образом, по мере движения вниз по стеку пакет накапливает заголовки. На стороне получателя — по мере завершения работы каждый слой передает свою «полезную нагрузку» на уровень выше, и заголовки удаляются. К полезной нагрузке относятся заголовки верхних по отношению к текущему уровню. Концептуально обмен данными происходит между фрагментами, находящимися на одном уровне в стеке. Клиент HTTP общается по протоколу HTTP с сервером HTTP, слои TCP общаются по протоколу TCP, ну и т.д.

Движемся вверх по стеку

На нижних уровнях стека протоколов используются сообщения двоичного формата, которые лучше всего описываются диаграммами вроде показанной на рисунке. Выше, на уровне приложений, протокол часто уже текстовый. Например, запрос HTTP (Hypertext Transfer Protocol — протокол передачи гипертекста) может выглядеть так:

```
GET /greeting.txt HTTP/1.1
Host: www.example.com
а ответ — так:
HTTP/1.1 200 OK
Content-type: text/plain
Content-length: 14
Hello World!
```

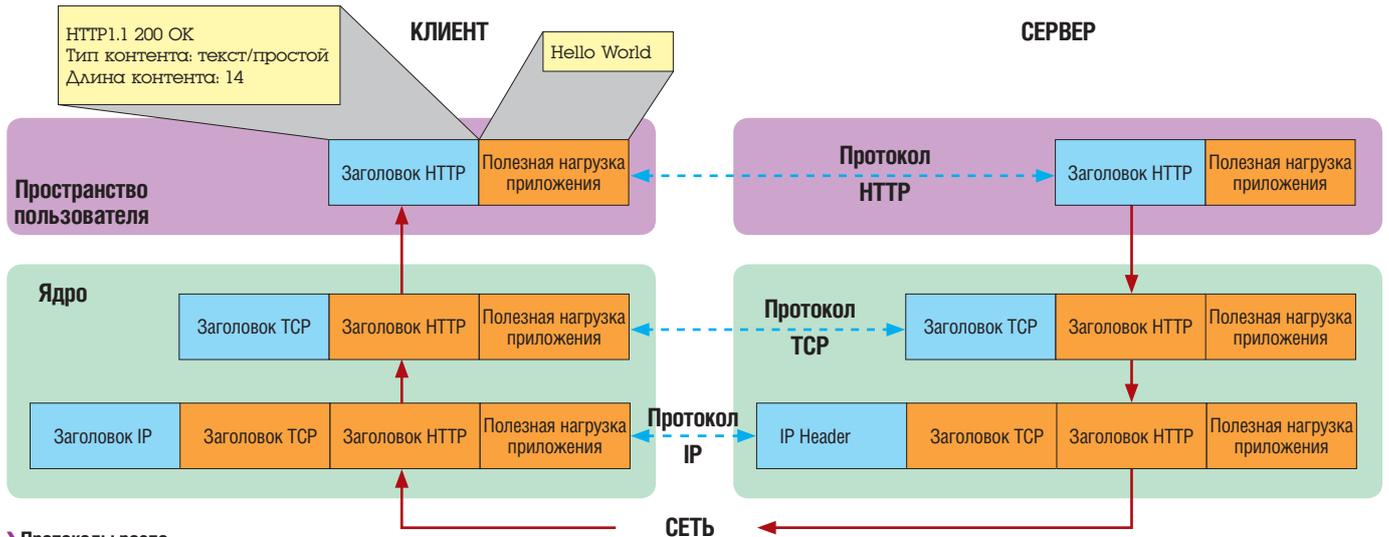
Даже на уровне приложений (по крайней мере, в этом случае) есть разделение между «заголовком» и «полезной нагрузкой». В приведенном выше HTTP-ответе первые три строки — это заголовок, а последняя — полезная нагрузка. Они разделяются пустой строкой — это часть протокола HTTP. »



Версия	ИHL (Длина заголовка Интернета)	Тип сервиса		Суммарная длина																											
Идентификация				Флаги				Смещение фрагмента																							
Время жизни				Протокол=6				Контрольная сумма заголовка																							
Адрес источника																															
Адрес получателя																															
Параметры																Дополнение															
Порт отправителя																Порт получателя															
Номер последовательности																															
Номер подтверждения																															
Смещение данных		Зарезервировано		U R G	A C K	P S H	R S T	S Y N	F I N	Окно																					
Контрольная сумма																Указатель срочности															
Параметры																Дополнение															
Данные																															

Заголовок IP
Заголовок TCP
Данные

» Традиционный способ документирования интернет-протоколов. Диаграмма имеет ширину 32 бита, и ее следует читать построчно.



➤ Протоколы располагаются по слоям. Заголовок каждого слоя (голубой) — часть полезной нагрузки слоя под ним (оранжевый).

Нечто абсолютное иное...

О протоколах достаточно. Что у нас с API? Мой верный Оксфордский словарь английского языка сообщает только, что это аббревиатура от “application programming interface [программный интерфейс приложений]”, но это мы и сами знаем. Обратившись к Википедии, мы найдем следующее: «программный интерфейс приложений (API) определяет, как компоненты программ должны взаимодействовать друг с другом», и это немного расплывчатое определение; но далее отмечается, что «на практике API часто реализуется в форме библиотеки, содержащей спецификации процедур, структур данных, объектов классов и переменных».

Классический пример API — интерфейс системных вызовов к ядру Linux. Это набор функций, которые могут вызывать программы для доступа к сервисам ядра — для обращения к файлам и каталогам, запуска новых процессов, подключения к сетевым сервисам, создания каналов и т. д. Вот фрагмент кода на C, который удаляет все файлы из каталога:

```
DIR *dir;
struct dirent *info;
char *n;
dir = opendir(".");
while ((info = readdir(dir)) != NULL) {
    n = info->d_name;
    if (strcmp(n, ".") && strcmp(n, "..")) {
        printf("removing %s\n", n);
        if (unlink(n) < 0) {
            perror(n);
            exit(2);
        }
    }
}
```

```
}
}
}
```

Если вы слабо знакомы с C, не беспокойтесь: просто обратите внимание на вызовы `opendir()`, `readdir()` и `unlink()`. Эти вызовы — часть низкоуровневого API управления каталогами в Linux.

API по своей природе зависят от языка, хотя часто встречаются «языковые привязки» к библиотеке для разнообразных языков. Для сравнения, вот эквивалентный код на Perl:

```
opendir(DIR, ".");
foreach $info ( readdir(DIR) ) {
    if ($info ne "." && $info ne "..") {
        if (unlink($info) != 1) {
            warn "$info: $!\n";
            exit(2);
        }
    }
}
```

Синтаксисы немного различаются, но вызовы функций `opendir()`, `readdir()` и `unlink()` видны ясно. Точно так же можно было написать это на Python или PHP; у каждого из них будет своя языковая привязка, но метод по сути останется тем же.

Замутим воду

Надеюсь, пока различие между протоколом и API понятно. Они никак не связаны друг с другом. Но здесь вода малость замутняется, так как обычно можно сворачивать протоколы (по крайней мере, протоколы уровня приложения) в библиотечные функции и в итоге получать API. Вот пример: RFC959 описывает почтенный протокол передачи файлов (FTP). Этот протокол «завернут» в библиотеку Python с подходящим названием `ftplib`. В этом модуле определен класс под названием `FTP`, который реализует клиентскую часть протокола FTP. Вот пример его использования:

```
>>> from ftplib import FTP
>>> ftp = FTP('ftp.debian.org')
>>> ftp.login()
'230 Login successful.'
>>> ftp.cwd('debian')
'250 Directory successfully changed.'
>>> ftp.retrlines('LIST')
-rw-rw-r-- 1 1176 1176 1066 Feb 15 09:23 README
-rw-rw-r-- 1 1176 1176 1290 Jun 26 2010 README.CD
... остальной вывод опущен ...
```

Опять же, не волнуйтесь, если вы не знакомы с Python; четко просматриваются вызовы `login()`, `cwd()` и `retrlines()`, которые

Запрос на комментарии

Самый первый RFC выпустил 7 апреля 1969 года Стив Крокер [Steve Crocker]. Крокер был одним из тех молодых студентов, которые еще в 1969 году по сути изобрели Интернет. В интервью журналу “Wired” Крокер объяснил, что они решили начать записывать некоторые из идей, приходивших им в голову, но ощутили, что «сам факт записывания наших разговоров можно рассмотреть как посягательство на власть, и кто-нибудь придет и наорет на нас...». С целью

подчеркнуть неофициальную природу этих записей и было выбрано название «Запрос на комментарии». Крокер говорит: «Я искренне полагал, что ко времени появления Сети уже будет некая формальная документация, и наши записки выбросят за ненадобностью как устаревшие. Но они уцелели. И этот стиль стал первичным при документировании».

На данный момент существует около 7000 RFC; все они доступны на сайте www.rfc-editor.org.

образуют часть API FTP. (Полный код см. на <http://bit.ly/ftplibclient>) Существует похожий модуль `http.client`, где определен класс `HTTPConnection`; его методы реализуют клиентскую часть протокола HTTP. Я выбрал Python для своих примеров (почти случайно; существуют библиотеки, предоставляющие привязки к FTP и HTTP для многих других языков).

А потом были web-сервисы

Если рассмотреть web-сервисы, то грань между протоколом и API еще более размывается. Мое представление о web-сервисах слегка перевернутое (некоторые сказали бы — старомодное), так как для меня это набор удаленных вызовов процедур. Основная идея в том, что некоторые процедуры (функции, подпрограммы, методы — называйте как хотите) сидят на удаленном компьютере, но их можно вызывать так, как если бы они были локальными. Процедуры, которые мы вызываем на самом деле, представляют собой лишь прокси. Они принимают входные параметры (3), сериализуют их в некоем хорошо определенном формате (4) и отправляют их на сервер (5), где они извлекаются из потока данных и передаются реальной процедуре (6). Эта процедура делает свою работу и (предположительно) возвращает результат (7), который снова сериализуется (8) и снова отправляется прокси на клиенте (9), где данные извлекаются из потока и возвращаются процедуре, вызванной изначально (10). (Приведенные цифры соответствуют нумерации на рисунке.)

Итак, здесь у нас есть и API (определяемое набором экспортируемых методов сервиса), и протокол (определяющий формат сообщений, передаваемых между клиентом и сервером при вызове этих методов).

Вот небольшой пример web-сервиса. Надеюсь, никого не обидит то, что он написан на C#.NET. Ядро этого сервиса — крошечный метод, который просто возвращает строковое представление текущего времени и даты:

```
[WebMethod]
public string RightNow() {
    return DateTime.Now.ToString();
}
```

Когда я «публикую» этот сервис, он размещается на web-сервере (в данном случае, IIS). Почему на web-сервере? Потому что (как мы увидим через минуту) протокол, используемый для вызовов процедур, расположен в стеке протоколов над HTTP, и если рассматривать большую часть стека, вызов процедуры выглядит просто как загрузка web-страницы. Сервер также предоставляет описание сервиса на языке WSDL (Web Service Description Language — язык описания web-сервиса), основанном на XML. На рисунке отмечены основные части этого описания. Видно, что описание пугающе подробное, как и сам XML. По сути же оно говорит только, что у сервиса есть процедура `RightNow`, которая не принимает входных параметров и возвращает строку. Обратите внимание, что это описание на WSDL не привязано к конкретному языку программирования или операционной системе.

Данный пример позволяет мне спросить: это API или протокол? На самом деле — ни то, ни другое: это описание сервиса. Но по сути оно определяет API как набор именованных операций с входными и выходными параметрами. Существуют средства для написания «прокси-методов», которые сериализуют передаваемые методу параметры и десериализуют возвращаемое значение (см. шаги 1 и 2 на рисунке). Для этого обмена данными используется вышеупомянутый протокол SOAP — он представляет собой XML-код, передаваемый в качестве полезной нагрузки запроса или ответа HTTP, которые, как мы уже видели, в свою очередь передаются по уровням протоколов TCP и IP.

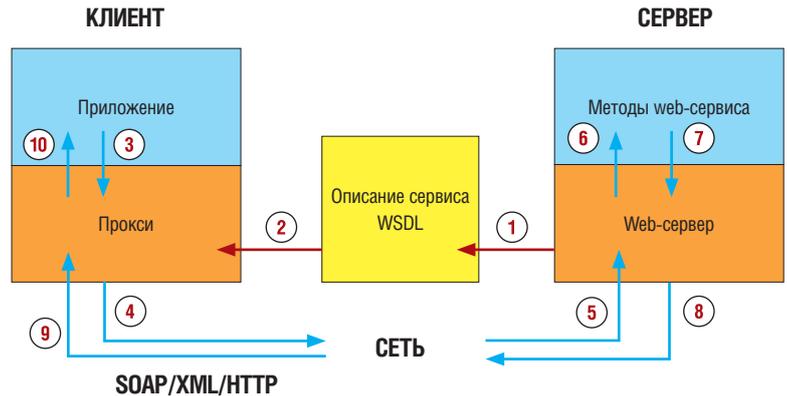
Думаю, главное здесь в том, что вокруг протокола вы всегда можете обернуть API, написав библиотечные процедуры, которые знают, как общаться по этому протоколу. Это и есть прокси.

Клиенты и серверы

Протоколы определяют общие правила обмена данными между клиентом и сервером; но как различить, кто из них кто? Предлагаю вам мое базовое определение этих объектов.

Клиент — это активный участник обмена, который ищет сервер, подключается

к нему и запрашивает его сервисы, тогда как сервер — участник пассивный. Задача сервера — связать свой «хорошо известный» порт с сокетом, перевести сокет в режим прослушивания подключений, откинуться в кресле и ждать, пока все начнется.



Разумеется, мой пример — всего лишь учебный, но в мире существует множество настоящих описаний сервисов. Например, в API web-сервисов Amazon EC2 определено более 300 операций, по поддержке создания виртуальных машин, жестких дисков, пар ключей и многого другого. На основе этих операций создаются утилиты командной строки (с именами вроде `ec2-create-keypair`, `ec2-run-instances` и `ec2-attach-volume`; они — немногим более, чем просто обертки вокруг тех 300 операций). На основе данного API также построена консоль управления AWS с web-интерфейсом. Похожую историю мы наблюдаем в API управления OpenStack.

Вот так. Как сказал бы Дуглас Адамс [Douglas Adams]: API почти, но все-таки не совсем, абсолютно непохожи на протоколы. LXF

» Для описания сервисов и сериализации параметров при вызове их методов в web-сервисах используются языки на основе XML.

```
<wsdl:types>
  <xs:schema elementFormDefault="qualified"
    targetNamespace="http://example.com/">
    <xs:element name="RightNow">
      <xs:complexType/>
    </xs:element>
    <xs:element name="RightNowResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1"
            name="RightNowResult" type="s:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</wsdl:types>

<wsdl:portType name="ClockSoap">
  <wsdl:operation name="RightNow">
    <wsdl:input message="tns:RightNowSoapIn"/>
    <wsdl:output message="tns:RightNowSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
```

» Жутко подробный WSDL используется для формального описания сервиса. И это только часть описания!

SOAP подпирает REST

Пока вы не написали на меня жалобу, спешу сказать, что я знаю: WSDL и SOAP — не единственные пацаны во дворе по части определения сетевых сервисов и обращения к ним. Есть еще REST, что расшифровывается малоинформативным «Передача

репрезентативного состояния [Representational State Transfer]». Не буду вызывать огонь на себя, отдав предпочтение одному из них, но REST определенно проще и, похоже, в последние годы стал гораздо популярнее SOAP.

ДИСТРИБУТИВЫ GNU/LINUX НА USB FLASH



Linux Mint 16



Ubuntu 14.04

А ТАКЖЕ

версии для юриди-
ческих лиц —
с лицензионным
договором
присоединения

USB Flash 8 Gb

495

рублей

www.linuxcenter.ru/shop/distros/usb/

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Свободные лекции

Звание свободного человека дороже всего.
Эврипид

Кафедра теоретической физики физического факультета Новосибирского государственного университета, а точнее, молодые и активные её представители организовали запись лекций её зрелых лидеров. В принципе это вполне себе рядовое явление даже в наших не шибко избалованных техникой университетах. Но было решено, что исходники лекций выложат в открытый доступ. Предложение использовать для видеофайлов лицензию CC-BY-SA (свободная, атрибутивная, с требованием распространения производных на условиях этой же лицензии) было принято без возражений, так как для науки свобода распространения информации является совершенно естественным состоянием.

Да, технически процедура публичного представления исходников получилась не очень удачной: использовалась не самая удобная площадка, и слэшдот-эффект всё сильно усугубил. Но со временем и размер видеофайлов уменьшится после перекодирования, и «торрент» с лекциями будет поднят, и на YouTube можно будет посмотреть, как В. С. Фадин, В. Л. Черняк, О. В. Жиров и т. д. несут в этот мир доброе (сильное и электрослабое взаимодействия, квантовая электродинамика, нелинейная и хаотическая динамика и т. д.) и вечное (как минимум ближайшие 10 лет) знание. Уверенность в том даёт именно свободная лицензия. Что тоже важно.
E.M.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Сжимать файлы 62

Нейл Ботвик продолжает свое руководство по главным командным утилитам рассказом про архивирование файлов посредством *Tar*.



Получать помощь 64

Застраили на проблеме с Linux? **Нейл Ботвик** покажет, что помощь ближе, чем вы думали, благодаря map-страницам.



Писать скрипты 66

Ирина Чернова решает автоматизировать все на своем пути, написав скриптов оболочки. Шансы безусловно есть!



Печатать 3D-объекты 70

Смастерите элегантный корпус к своему Raspberry Pi, применив *FreeCAD*, 3D-принтер, код Python и советы **Робина де Йонга**.



Собирать облака 74

Джолион Браун использует «упертый скрипт» *DevStack* для быстрого развертывания собственного облака *OpenStack* на одиночной виртуальной машине.



Вести заметки 78

Две открытые программы, *NixNote* и *Etherpad*, помогают **Марко Фиоретти** обзавестись Evernote-подобной системой для совместной работы онлайн.

АКАДЕМИЯ КОДИНГА



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

Python

Стартуем с нуля к основам программирования, с **Даниэлем Сэмюэлсом**. С. 82

Python

Нейл Ботвик создает пути и каталоги, работающие невзирая на ОС. С. 84

Python

Про все, что надо знать для хакинга в *Minecraft Pi*, расскажет **Джонни Бидвелл**. С. 86, 88

MySQL

Лихо перетаскиваем электронные таблицы в базу данных и обратно, как **Кент Ельчук**. С. 92

Tar: Архивируем свои файлы

Архивирование таит в себе много загадок, таких как *Zip* и *Tar*, но наш эксперт Нейл Ботвик приоткроет завесу тайны над ними.



Наш эксперт

У Нейла Ботвика по компьютеру в каждой комнате, но по соображениям безопасности он ничем не раскajat, где находится центральный сервер.

Большинство читателей этого журнала наверняка хорошо знакомы с созданием, отправкой и получением ZIP-файлов. *Zip* получает набор файлов и сохраняет их в ZIP-архиве, сжимая данные в процессе создания архива. Наряду с содержимым файлов *Zip* также сохраняет все их метаданные, т.е. дополнительную информацию, связанную с объектом. Для файлов она включает время изменения, владельца и права доступа и, конечно, имя каждого файла. При распаковке архива вся эта информация извлекается, и исходный набор файлов воссоздается в первоначальном виде. Это очень удобно!

Архивы используются по-разному, но чаще всего — для связывания нескольких файлов при загрузке (с одним файлом удобнее работать, а благодаря сжатию он загрузится быстрее), ну и для создания резервных копий. Формат *Zip* возник давно, поэтому им могут пользоваться все, и все ОС его поддерживают. Тем не менее, он не лишен недостатков. Главный недостаток — плохое, по нынешним меркам, сжатие. За последние 25 лет технологии сжатия изрядно продвинулись, и хотя *Zip* отчасти усовершенствовали, появились альтернативы получше. Еще один недостаток *Zip* в том, что данные архивируются в файл, тогда как иногда нужно переслать на другое устройство или сервис сами данные.

Стандартная программа архивации для Unix-подобных ОС — *Tar*, названная так потому, что первоначально использовалась для создания архивов на ленточных накопителях (Tape Archive). Он работает не так, как *Zip*, потому что все архивируемые данные

отправляет в стандартный вывод и по умолчанию не сжимает их, поскольку во многих ленточных накопителях уже есть встроенное аппаратное сжатие. Отсутствие режима сжатия может показаться недостатком, но на самом деле это преимущество. Раз уж *Tar* может направить свои данные во внешнюю программу сжатия, вы можете применить любую программу сжатия, которая вам нравится — даже такую, которой не существовало во времена создания *Tar*.

Программы сжатия получают один файл или поток данных и формируют один сжатый файл или поток, поэтому процесс разбивается на две части: архивирование и сжатие. Хотя это может показаться усложнением, *Tar* прекрасно позаботится о деталях сам. Пусть у нас есть каталог **foo**. Мы хотим создать из него архив. Это можно сделать одним из следующих способов:

```
tar cf foo.tar foo
tar czf foo.tar.gz foo
tar cjf foo.tar.bz2 foo
tar cJf foo.tar.xz foo
```

Параметр **c** сообщает программе *Tar*, что мы создаем архив, а **f** — что архив нужно сохранить в файле с указанным именем. Следовательно, первая команда создает несжатый архив в файле **foo.tar**. В последующих командах появляется дополнительный аргумент, который сообщает *Tar*, какой именно тип сжатия использовать: **z** соответствует *gzip*, **j** — *bzip2*, а **J** — *xz*. (Обращайте внимание на регистр букв!)

Каналы и потоки

Краеугольный камень философии Unix-подобных операционных систем в том, что утилиты должны быть специализированными: каждая утилита должна решать одну задачу, и делать это хорошо. Следовательно, нужен способ связать программы в цепочку для решения более сложных задач, и для этого используются каналы. У каждой консольной программы есть стандартные потоки ввода, вывода и ошибок (обычно обозначаемые `stdin`, `stdout` и `stderr` соответственно). Через стандартный поток ввода программа получает информацию, и по умолчанию, если программа запущена в терминале, это клавиатура. Стандартный вывод — это вывод команды, который по умолчанию отображается в окне терминала. Стандартный поток ошибок отправляется туда же, но все потоки можно перенаправить в другие места. Взгляните на пример:

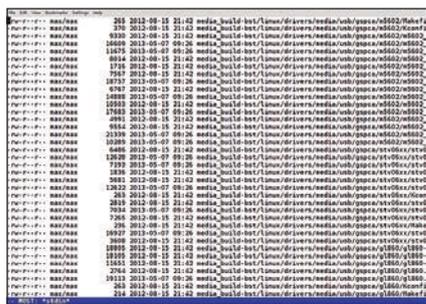
```
tar c foo | sdc >foo.tar.sdc
```

Символ **|** означает «канал», и он подключает стандартный вывод одной программы к стандартному вводу другой. Таким образом, *Tar* архивирует содержимое **foo** и отправляет его в `stdout`, который затем перенаправляется на вход `sdc`. Большинство программ сжатия по умолчанию работают с файлами:

```
gzip somefile
```

Команда сжимает файл в **somefile.gz**. Если файл не указан, то сжимается стандартный ввод, а результат направляется в стандартный вывод. Символ **>** выполняет перенаправление и выводит стандартный поток из терминала в файл; таким образом, у нас получается архив **foo.tar.sdc**.

Стандартный поток ошибок перенаправить с помощью символа **>** нельзя. Если что-то идет не так, сообщения об ошибках выводятся в терминал — если бы они отправлялись в файл, вы бы так и остались в неведении.



» При запуске с параметром **t** (**test** — проверка) *Tar* не только проверяет, что содержимое архива корректно, но и отображает полное содержимое архива — обычно его достаточно пропустить через утилиту постраничного просмотра вроде **Less**.

Типы сжатия

Существует два типа сжатия: с потерями данных и без потерь данных. При сжатии с потерями данных можно добиться большего сжатия, осторожно удаляя те данные, которые существенно не влияют на результат. Сжатие с потерями используется в файлах JPEG и MP3. При архивировании используется сжатие без потерь, и при распаковке вы получаете все тот же исходный объект. Существует несколько методов сжатия без потерь — вот несколько наиболее распространенных:

- » **Deflate** Этот метод сжатия по умолчанию используется в *Zip*; он старый и не слишком эффективный, но быстрый и хорошо поддерживается.
- » **Compress** Старая программа сжатия для Unix. У файлов, сжатых с ее помощью, расширение *.Z*, но в дикой природе вы их вряд ли встретите. С ней также связаны патентные проблемы.

» **Gzip** Замена *Compress*, полностью открытая и используемая по сей день. В *Gzip* используется *Deflate*, но размер получаемого архива обычно меньше, чем у *Zip* с настройками по умолчанию. Она не дает наилучшего сжатия, но быстро сжимает и распаковывает файлы, что делает ее хорошим вариантом, когда размер файла не слишком важен.

» **Bzip2** Более эффективная программа сжатия, но эффективность достигается за счет скорости. Особенно замедляется сжатие, но размер получаемых файлов будет меньше.

» **xz** В этой, самой свежей программе используется алгоритм сжатия LZMA2, также применяемый в *7-Zip*. Она работает быстро — особенно при декомпрессии — и дает отличные результаты. Эта программа сжатия используется во многих ключевых проектах Linux, включая *Coreutils* и само ядро.

```
[nelz@hactar TerminalTutorials/Part4] % ls test.tar.*
-rw-r--r-- 1 nelz users 5.6M Apr  4 12:17 test.tar.bz2
-rw-r--r-- 1 nelz users 8.2M Apr  4 12:17 test.tar.gz
-rw-r--r-- 1 nelz users 5.4M Apr  4 12:17 test.tar.xz
[nelz@hactar TerminalTutorials/Part4] %
[nelz@hactar TerminalTutorials/Part4] % gzip -l test.tar.gz
compressed      uncompressed  ratio uncompressed_name
8358353         33525760    74.5% test.tar
[nelz@hactar TerminalTutorials/Part4] %
[nelz@hactar TerminalTutorials/Part4] % xz -l test.tar.xz
Strms  Blocks  Compressed Uncompressed Ratio Check Filename
1      1      5,565.8 KiB  32.0 MiB  0.168 CRC64 test.tar.xz
[nelz@hactar TerminalTutorials/Part4] %
```

» Один и тот же архив, сжатый разными программами. Версия *Gzip* заметно больше, но быстрее сжимается и распаковывается.

У этих аргументов есть еще и длинные варианты, которые делают команды более читаемыми, но большинство из нас ленивы и пользуются короткими: их быстрее набирать. Однако при желании команда может быть и такой:

```
tar --create --gzip --file foo.tar.gz foo
```

Расширение файла указывать не обязательно, но лучше все же указать — так можно легко увидеть тип создаваемого архива. Система определит тип архива сама, помощь ей для этого не требуется. Для распаковки архива достаточно заменить *c* на *x* (или *--create* на *--extract*). Учтите, что при распаковке не нужно указывать тип сжатия — *Tar* определит его сама:

```
tar xf foo.tar.gz
```

Другой параметр, который может вам пригодиться — *v* или *--verbose*; с ним *Tar* будет сообщать вам, что она делает.

Если у вас есть готовый архив, может возникнуть желание посмотреть, что в нем содержится, не распаковывая архив. А если вы создали архив, особенно резервную копию, сами, естественно будет пожелать проверить, что он создан без ошибок. С параметром *test (t)* *Tar* проверяет целостность и выводит содержимое архива.

```
tar tvf foo.tar.gz
```

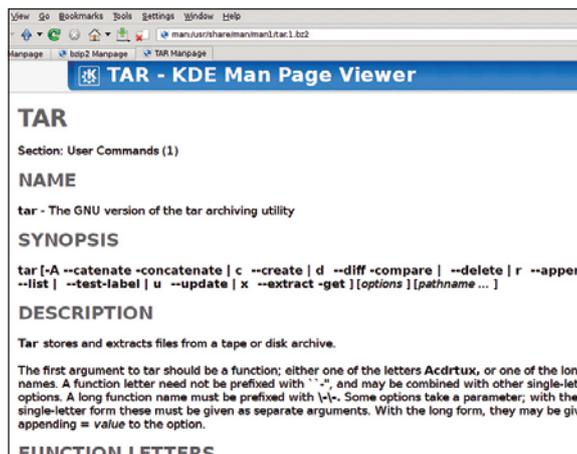
Это основные параметры *Tar*, но на самом деле их гораздо больше: например, есть *A* или *--concatenate* для добавления файла в существующий архив, вместо создания нового.

Смотрим в будущее

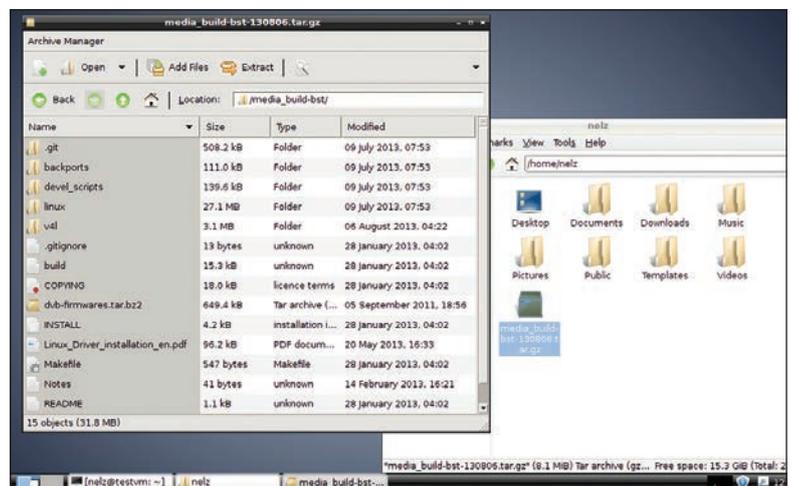
Мы сказали, что *Tar* может поддерживать любой новый формат сжатия, так как передает сжатие другой программе. Есть параметры, которые сделают это автоматически для *gzip*, *bzip2* и *xz*; ну, а вдруг кто-то придумает новую программу сжатия? Например, *sdcc* — супер-дупер-компрессор? В таком случае можно было бы создать несжатый архив, а затем сжать его с помощью *sdcc*; но это затратно по ресурсам, да и медленно. Вместо этого воспользуемся каналом:

```
tar c foo | sdcc >foo.tar.sdcc
unsdcc foo.tar.sdcc | tar xv
```

Здесь мы используем только параметр *--create*. Отсутствие места назначения вынуждает *Tar* отправить данные архива в стандартный вывод, который затем через канал направляется программе сжатия *sdcc*. Вторая команда делает все наоборот: производит декомпрессию архива и отправляет его *Tar* для распаковки. LXF



» У *Tar* и других программ сжатия есть man-страницы с описанием многочисленных параметров. Но обычно достаточно тех, что описаны в данной статье.



» Содержимое архива позволяет просмотреть большинство рабочих столов. Здесь показан результат двойного щелчка по *tar*-архиву в *LXDE* (рекомендуемый для Raspberry Pi).

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Man-страницы: Помощь близка

Нейл Ботвик роется в справочной литературе, встроенной в операционную систему Linux для того, чтобы утопающие помогли сами себе.



Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он нипочем не расскажет, где находится центральный сервер.

РTFM уже давно считается боевым ключом линуксоидов-экспертов специально чтоб отпугнуть новичков. Если вы раньше не сталкивались с этим термином — он означает нечто вроде 'Read The Fine Manual [Прочитайте Это Прекрасное Руководство]' [иронически, — прим. пер.]. Наверное, легко представить досаду человека, которого в сотый раз спрашивают про одно и то же, причем ясно описанное в руководстве. Однако прочитать документацию по программе пользователю удастся лишь в том случае, если он сумеет её найти. К счастью, справочные материалы в Linux таки есть; давайте же поговорим о каждом.

Еще до поиска руководства вспомните, что во многих программах встроена справка. Запустите соответствующую программу команду в терминале с параметром `--help`, чтобы увидеть все доступные опции. Это касается и графических программ, и команд оболочки. Например:

`firefox --help`

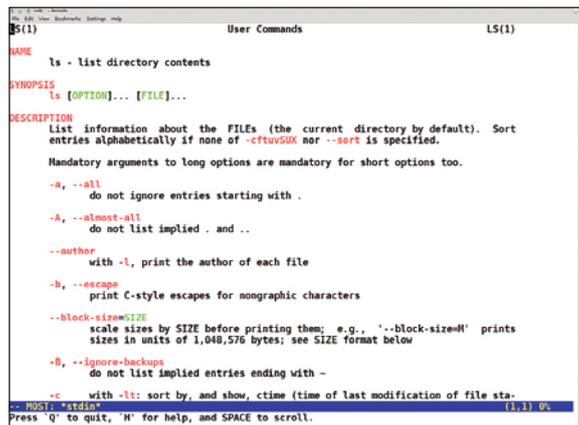
Если вам нужны детали, значит, и пришла пора RTFM, что в системах Linux обычно означает обращение к man-страницам. Там вы найдёте информацию обо всём, что установлено в вашей системе, и просмотреть её можно, введя `man` в командной строке. Если вам интересно, как именно это работает — следуя классическому примеру рекурсии, откройте терминал и наберите

`man man`

Man-страница — это одиночная страница со справочной документацией по заданной теме. Команда `man` представляет данную информацию в читаемом виде и отображает её с помощью встроенного скрипта постраничного вывода — как правило, `less`. Это означает, что вы используете навигацию `less`: клавиши управления курсором для прокрутки вверх и вниз, пробел для пролистывания и так далее. Man-страницы бывают очень длинными, поэтому попробуйте воспользоваться поиском с `man bash`. В `less`, введите `/`, чтобы искать далее (или `?` для поиска снизу вверх), и искомое выражение. Далее используйте `n` для перехода к следующему совпадению или `N` для возврата к предыдущему. Man-страницы содержат следующие разделы:

- 1 Команды пользователя
- 2 Системные вызовы
- 3 Библиотечные функции C
- 4 Устройства и специальные файлы
- 5 Форматы файлов и соглашения
- 6 Игры и прочее
- 7 Разное
- 8 Инструменты системного администрирования и демоны

Как обычному пользователю, вам по-прежнему понадобится только разделы 1, 5 и 8 (ну и, может быть, 6). Если вместе с командой `man` вы укажете номер раздела, то сможете просматривать его отдельно; в противном случае вам покажут первое найденное совпадение. Это необходимо, поскольку страниц



» Пользователи KDE могут читать страницы `info` и `man` в браузере, с активными ссылками, благодаря труженикам `KIO slaves`.

Просмотр на рабочем столе

Man и `info` предназначены для чтения в терминале, с учётом того, что вам, возможно, потребуется использовать их на системе без рабочего стола. Но существуют графические утилиты для их просмотра; наиболее удобные — в KDE, где вы можете нажать `Alt+F2`, ввести `man:command` или `info:command` и получить HTML-версию документа для просмотра в `Konqueror`. Есть

также утилиты `tkInfo` и `tkMan` для просмотра соответствующих форматов в графическом интерфейсе.

Существует несколько сайтов, содержащих полные коллекции man-страниц, например, www.linuxmanpages.com и <http://manpages.ubuntu.com>. Они особенно полезны, если вы хотите прочитать о чём-то у вас не установленном.



» Вкладка `Preview` со точками. находится курсор мыши.

с одинаковым именем может быть несколько. Команда **passwd** используется для определения пользовательских паролей, которые хранятся в файле **/etc/passwd**. Попробуйте:

```
man passwd
man 1 passwd
man 5 passwd
```

Первые два содержат команду **passwd** из раздела 1, а третий показывает справочную страницу для файла паролей. Это одно из преимуществ **man**-страниц — здесь зафиксировано всё: команды, файлы конфигурации, библиотеки функций и многое другое. Но не только конкретные команды или файлы: раздел 7 содержит справочные страницы для самых разных вещей. Хотите знать, как работают символические ссылки, или что происходит, когда вы включаете компьютер? Попробуйте

```
man 7 symlink
man 7 boot
```

Скорая помощь

Man — это нечто большее, чем стопка отформатированных страниц текста и программа для их просмотра. **Man** содержит базу данных с возможностью поиска страниц, автоматически обновляемую через **Cron**, а также некоторые другие программы для работы с ней. На каждой **man**-странице есть раздел **NAME**, включающий краткое описание её содержания. Команда **whatis** даёт вам это описание — объясняет, что это за программа (или файл), не давая детальных опций.

Вот классический, любимый технарями, рекурсивный пример:

```
whatis whatis
```

```
whatis (1) - search the whatis database for complete words
```

А вот более быстрый способ увидеть действия команд, особенно если вас интересует не одна, а больше:

```
whatis grep sed symlink
```

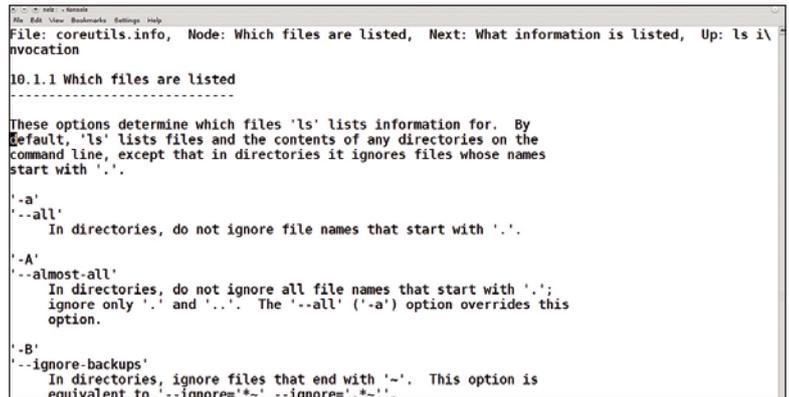
Команда **whatis** ищет только по имени, и только целые слова, поэтому предполагается, что название команды вам известно и вы просто хотите знать, зачем она нужна. Для более широкого поиска используйте **apropos**, которая выполняет аналогичную работу, но ищет по описаниям, а также и возвращает все совпадения — сравните эти две команды:

```
whatis png
```

```
apropos png
```

Существует еще одна любимая GNU-проектом форма документации: **info**-страницы. Если **man**-страница, по сути, представляет собой один очень длинный текстовый файл с намёком на форматирование и разметку, то документ **info** уже содержит дерево страниц, связанных в один файл. Он напоминает скорее HTML, чем простой текст, но предназначен для чтения в текстовой консоли, и все «страницы» содержатся в одном файле. Как легко догадаться, команда для чтения **info**-страниц будет выглядеть следующим образом:

```
info info
```



И на сей раз, эта самореференция — не ради забавы. На **man**-страницах можно ориентироваться интуитивно — так же просто, как читать любой другой текст в подобном скрипте. **Info** использует другой набор основных команд, поэтому чтение лучше начать с собственной страницы **info** для **info**. Навигация по страницам осуществляется как обычно, но если вы нажмете **Enter**, наведя курсор на ссылку (отмеченную звездочкой), то перейдете в этот узел. Для возврата нажмите клавишу **u**. Можно также переходить по уровням дерева документации, используя **n** и **p**, которые доставят вас к следующему или предыдущему из узлов. Если вам случилось видеть, как выглядит документация GNU в Интернете (<http://bit.ly/grubmanual>), то с этой схемой вы уже знакомы. Она проще, чем HTML, и навигация осуществляется, как правило, в пределах одного уровня или же перемещением на уровень вверх и вниз.

Вы спросите, в чем смысл этой структуры? Ну, если вы когда-либо пытались найти какую-либо информацию на длинной **man**-странице, например, **man bash** или **man mplayer**, то представляете, сколько времени и сил порой тратится при этом подходе «все на одной странице». **Info**-документы делятся на разделы и главы, представляя данные в более ясной и сжатой форме. Большинство программ GNU имеют довольно короткие **man**-страницы и более детальное описание на страницах **info**. Разбивка документа на страницы меняет сам способ поиска. Введите **s**, затем искомое выражение, и нажмите **Enter**. Вы увидите информацию по следующему совпадению, даже если оно находится на другом узле. Продолжайте нажимать **s** и **Enter**, не меняя критерий поиска, чтобы увидеть дальнейшие совпадения. С помощью этих клавиш, плюс **q**, для выхода, вы сможете с легкостью ориентироваться на **info**-страницах. Вероятно, вы удивлены, почему мы не используем HTML. Главным образом, потому, что **info** появился раньше HTML, и документы в формате **info** хранятся в одном файле. Концептуально же они похожи настолько, что можно воспользоваться **info2html** (<http://info2html.sourceforge.net>) для преобразования **info**-документа в несколько HTML-страниц. **ИЗБ**

➤ Страница **ls info** даёт более подробное описание параметров группы с учётом функции. **Info**-страницы обычно более подробны, чем **man**-страницы.

Распечатываем руководство

Иногда удобным бывает иметь под рукой печатный вариант справочного руководства. Поскольку **man**-страницы хранятся в формате разметки (а при отображении конвертируются программой **man**), вам понадобится воспользоваться опцией **-t**, чтобы преобразовать их в формат,

пригодный для распечатки (по умолчанию — Postscript), например:

```
man -t somecommand | lpr
```

Postscript — это вссывод **output** на **stdout**; перенаправление его в **lpr** пошлет вывод прямо на принтер. Также можно преобразовать Postscript в PDF и создавать

версии **man**-страниц, которые можно читать с планшетов или электронных книг:

```
man -t somecommand | ps2pdf - somecommand.pdf
```

Документы **info** распечатываются командой **col**:

```
info somecommand | col -b | lpr
```

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



Linux: Скрипты автоматизируют

Из-за сложностей с пониманием графических интерфейсов Ирине Черновой пришлось выучить наизусть все команды оболочки...



Наш эксперт

Ирина Чернова мечтает попасть в тот день, когда число пользователей свободно-го ПО превысит число пользователей коммерческого. Надеется, что для этого ей не понадобится машина времени.



Большинство пользователей ПК (а таковое, к сожалению, составлено из пользователей Windows) не мыслит себя вне графического интерфейса типа «наведи-и-щелкни» (GUI). Однако любой человек, которому приходится длительно работать за компьютером в силу своей профессиональной (да и любительской) деятельности, сталкивается с необходимостью выполнять десятки или даже сотни однотипных операций, отнимающих значительную часть его времени. Графический интерфейс здесь не помощник. Но в Linux, как и следовало ожидать,

имеется великолепное встроенное средство для автоматизации любых процессов: язык сценариев (скриптов) командной оболочки — Shell Script.

На языке оболочки можно написать простенькую программу, а то и прототип серьезного приложения, причем для этого не потребуется устанавливать дополнительное программное обеспечение (интерпретатор скриптов оболочки входит в состав каждого дистрибутива Linux) или тратить много времени на самообразование. Синтаксис этого языка минималистичен и прост в освоении — особенно для людей, который имеют опыт программирования на каком-либо C-подобном языке. Философией программ мира UNIX/Linux является «выполнять только одну задачу, но делать это хорошо» — а далее составлять сложные задачи из ряда более мелких. Поступая истинно в этом духе, мы можем собрать в скрипте любую комбинацию команд оболочки, которая выполняет последовательность необходимых нам действий, и проделывать эти действия всего лишь одним вызовом скрипта.

Наш урок представляет собой краткое введение в язык скриптов. Мы собираемся приоткрыть читателю невероятные возможности скриптов оболочки и пробудить в нем интерес к дальнейшему изучению этой технологии.

Привет, мир!

По традиции, при изучении какого-либо языка программирования первым делом полагается написать простейшую программу, которая выводит на экран приветственную фразу «Hello world [Здравствуй, мир]!». Не будем от этой традиции отступать. Запишем в текстовом файле следующие строки:

```
#!/bin/bash
echo "Hello world"
```

Так наша программа будет выглядеть на языке оболочки. Если вы захотите снабдить ее комментариями, вставляйте их в виде строк, начинающихся со знака # (диез):

```
# Комментарий обозначается знаком диеза
```

Интерпретатор будет такие строки игнорировать, а вам они помогут разбираться в позабытом ходе своих мыслей, особенно когда ваши скрипты станут сложнее.

Наш код необходимо сохранить как файл с расширением .sh — дадим ему имя **myfirstscript.sh**. Итак, мы создали свой первый скрипт! Но работа еще не завершена: чтобы скрипт можно было запускать вызовом по имени, наш файл следует превратить в исполняемый. Для этого зайдите в терминал и наберите одну из следующих команд:

```
chmod +rx myfirstscript.sh
chmod u+rx myfirstscript.sh
```

Немного истории

Командная оболочка Bourne Shell (она же — *bash*) появилась в конце 1970-х годов сразу после того, как UNIX был переписан с Assembler'a на C, и с тех пор развивается вместе с операционной системой. Одновременно возник и язык скриптов.

Они различаются выдачей прав на исполнение файла: первая решает запускать скрипт любому пользователю в системе, а вторая — только владельцу скрипта.

Вот теперь наш код превратился в команду, запускаемую в терминале: зайдите в папку, куда его записали, и наберите

```
./myfirstscript
```

(расширение набирать не обязательно). Или передайте его на исполнение интерпретатору:

```
bash myfirstscript
```

Что в шалаше

С командой **echo** все более или менее понятно; но что означает загадочная строка `#!/bin/bash`? С этой строки должен начинаться любой скрипт оболочки (далее будем везде ее подразумевать, а приводить только собственно команды). Последовательность символов `#!` называют `shebang` (по-английски это означает «шалаш» — не правда ли, визуальное сходство есть?); строка указывает системе, какой интерпретатор использовать для выполнения кода, поскольку их имеется немало количество. Здесь мы выбрали *bash*, который (что и указано) проживает в папке **bin** — там хранятся готовые к исполнению двоичные `[binary]` системные программы. Самые распространенные интерпретаторы — *bash* и *sh*. Поясним, чем они отличаются друг от друга.

» **Sh** Самый скромный по функциональности, а следовательно, и самый компактный интерпретатор. Содержится в любой UNIX-системе. Если вы собираетесь портировать свой скрипт на устройства, современность конфигурации которых вызывает сомнения, пишите в первой строке `#!/bin/sh`.

» **Bash** Любой скрипт, выполняемый в *sh*, непременно выполнится и в *bash*. Но обратное утверждение неверно: *bash* содержит значительное число дополнительных особенностей — локальные переменные, проверка почтовых ящиков, подсветка синтаксиса, работа с учетными записями пользователей и т. д. Входит в состав практически всех дистрибутивов Linux.

Интерпретаторы *csh* и *ksh* по широте функциональности обгоняют *sh*, но уступают *bash*. А *tcsh*, *zsh*, *rc*, *es* имеют функции, которых нет в *bash* — это модификации *bash*, применяемые в особых ситуациях (выходящих за рамки этой статьи).

Между прочим, точно так же в терминале запускаются скрипты, написанные на других скриптовых языках программирования (Perl, PHP, Python и т. д.), были бы в системе соответствующие интерпретаторы:

```
perl perlscript.pl
```

```
php file.php
```

```
python script.py
```

Меры предосторожности

Однако вернемся к оболочке. Во время обучения программированию на языке скриптов оболочки лучше быть авторизованным в системе не как администратор, а как простой пользователь, которому не дозволено вторгаться в критически важные области системы. В противном случае, ошибка в названии утилиты, неверный путь к файлу или бесконечный цикл имеют шанс слегка подпортить вам жизнь.

Перед запуском скрипта стоит тщательно проверить код на синтаксические ошибки и правильность указания имен команд и путей к файлам (это особенно важно, если скрипт подразумевает удаление чего-либо). Для поиска ошибок в скрипте необходимо запустить его в режиме отладки, с параметром `-x` (а если вы хотите видеть более подробную информацию об исполнении скрипта, припишите параметр `-v`):

```
bash -xv myfirstscript
```

После строки с `shebang` не помешает вставить строку

```
set -o errexit
```

— она не даст скрипту продолжить работу, если какая-либо из команд в нем была завершена аварийно.

Основы синтаксиса

В языке оболочки есть понятие переменных. Им можно присваивать значения и производить над ними различные операции. Переменные различаются по типам — например, `local` (для использования внутри функции); или можно назначить тип `readonly` (тогда получится константа, значение которой изменять нельзя). Пример:

```
readonly koff=1.3
```

```
echo "Введите число"
```

```
read D
```

```
echo D*koff
```

Приведённая программа просит пользователя ввести число, считывает введенное значение в переменную `D` и выводит на экран результат его умножения на заранее определенный коэффициент. Команде **echo**, как видите, предлагается вывести результат вычисления по приставленной к ней формуле. Улавливаете? Добавив еще переменных и подставив произвольную формулу, таким образом можно запросто запрограммировать собственный калькулятор для индивидуальных расчетов!

Кстати о формулах. Как любой приличный язык программирования, язык скриптов оболочки позволяет описывать функции. Это делается для того, чтобы единожды записанный код можно было применять многократно. Допустим, нам надо найти сумму квадратов нескольких чисел. Опишем функцию возведения в квадрат:

```
function square(x){
```

```
  x=x*x
```

```
  return x
```

```
}
```

```
echo square(2)+square(3)+square(4)
```

Данный код выведет на экран число 29. Сначала мы описали функцию умножения числа `x` само на себя, принимающую один параметр — тот самый `x`. Однако описание — оно и есть описание, и само по себе ничего не делает. Заставит функцию работать ее «вызов», то есть упоминание ее по имени с указанием конкретного значения параметра, что мы и сделали трижды. Понятно, что приведенный пример несколько искусственный (в конце концов, набирать `square(2)` ничуть не быстрее, чем просто `2*2`); но если требуется вычисление функции значительно большей сложности, вы сможете здорово сэкономить свои труды.

Для повторяющихся действий предусмотрены, конечно, и операторы цикла. Попробуем поочередно выводить на печать элементы списка, он же «массив».

```
workernames=(“Иванов”,“Петров”,“Сидоров”)
```

— это описание в скрипте массива из текстовых строк, а вот и «активная» часть соответствующего скрипта:

```
for workername in workernames
```

```
do
```

```
  echo $workername
```

```
done
```

Мы прошлись в цикле по всем элементам.

Рассмотрим пример цикла похитрее. Произведем поэлементный обход содержимого папки, найдем в ней все файлы *Excel* и конвертируем их в CSV (с помощью программы-утилиты *xls2csv*):

```
for xls in /path/*.xls
```



Справку по командам оболочки можно получить в любой удобный момент (даже не имея соединения с Интернетом) — стоит лишь ввести в командной строке **man bash** или **info bash**. Что означают эти команды — см. Учебники, стр. 64.

Полезные ссылки

Полную документацию по технологиям, упомянутым в ходе нашего урока, вы найдете в Сети.

- » **Shell Script** <http://linuxcommand.org>, <http://www.linux-commands-examples.com/>, http://www.opennet.ru/docs/RUS/bash_scripting_guide/
- » **mailutils** <http://mailutils.org>
- » **Libre Office API** <http://api.libreoffice.org>
- » **ImageMagic** <http://www.imagemagick.org/>
- » **Mozrepl** <https://github.com/bard/mozrepl>
- » **Imacros** <http://wiki.imacros.net/>
- » **Zenity** <http://live.gnome.org/Zenity>
- » **ffmpeg** <https://www.ffmpeg.org>
- » **gzip** <http://www.gzip.org>

Море теоретической и практической информации о системном администрировании с помощью скриптов размещается на <http://www.linuxsysadmintutorials.com>.

```
do
xls2csv "$xls" "${xls%.xls}.csv"
done
```

Запись *.xls означает «все файлы с расширением xls». А теперь представьте, сколько времени ушло бы на это в графическом интерфейсе... (Если зашла речь о конвертации файлов из одного формата в другой, то для этого в Linux есть целый ряд специальных утилит: *Convert*, *Unix2dos*, *Dos2unix* и т.п.) Данный скрипт уже замаскирован на обращение «изнутри себя» к сторонней утилите командной строки.

В языке скриптов оболочки также есть стандартные для всех популярных языков программирования операторы условия (синтаксис аналогичен языку C). Вот пример:

```
if ["$UID" -ne "$ROOT_UID"]
then
echo "Войдите в систему от имени root!"
else
echo "Добро пожаловать!"
fi
```

Здесь мы проверяем наличие у пользователя прав администратора, сравнивая его идентификатор (UID) с идентификатором суперпользователя-root (для этих идентификаторов предусмотрены встроенные переменные, помеченные знаком \$). Если идентификаторы не равны, значит, пользователь администратором не является, и его попросят выйти вон.

Смотрите: в зависимости от текущей ситуации, вы можете реагировать на происходящее по-разному, организовав настоящие интерактивные сценарии!

Работа с приложениями

Чем дальше — тем интереснее. Из скриптов можно запускать не только простенькие утилиты, но и привычные вам графические приложения. Возьмем, скажем, *Libre Office*: почему бы не автоматизировать работу с офисными документами? Для этого воспользуемся программным интерфейсом — API — *Libre Office*. Через него вы можете поработать со следующими типами документов:

- » текстовый документ формата ODT (-writer);
- » электронная таблица (-calc);
- » файл векторной графики (-draw);
- » презентация (-impress);
- » HTML-страница (-web).

Скажем, чтобы конвертировать PDF-файл **myfile.pdf** в формат ODT и вывести его на печать, сделайте следующее:

```
libreoffice --headless --convert-to odt myfile.pdf
libreoffice -p myfile.odt
```

Работа в браузере

У многих людей значительная часть работ, которые они выполняют за компьютером, делается в браузере. Работу с сайтами и web-приложениями тоже легко автоматизировать. На это есть несколько способов.

Первый — установить дополнение MozRepl и обращаться посредством скриптов к его API. Например, откроем какой-нибудь сайт:

```
repl> getPageUrl()
http://www.site.com
```

Изменим заголовок вкладки на произвольный текст:

```
repl> document.title="Произвольный текст"
```

Второй вариант автоматизации действий в браузерах *Firefox*, *Safari* и *IE* (немного колдовства в *Wine*, и *IE* прекрасно запустится и в Linux) — использование дополнения *Imacros*. Это очень просто:

1 Установите через меню браузера (в *Firefox* — пункт Add-on в выпадающем списке Tools) дополнение *Imacros*.

2 Создайте с его помощью макрос (используя функцию записи или напрямую набивая код).

3 Выполняйте сохраненный макрос (файл в формате iim) командой

```
"путь к браузеру" imacros://run/?m=somemacros.iim
```

Функциональности MozRepl и *Imacros* идентичны — они позволяют с помощью программного кода имитировать действия пользователя, а именно:

- » щелчки по кнопкам и по точкам на экране с заданными координатами (x,y);
- » заполнение текстовых полей, выбор значений в выпадающих списках и т.п.;
- » переходы между вкладками;
- » и многое, многое другое.

Работа с почтой

Для автоматизации работы с почтой можно воспользоваться библиотекой *mailutils* (предварительно проверив, установлена ли она в системе и настроен ли доступ к почтовому серверу). Она содержит программы-демоны, реализующие протоколы доступа, программы доставки почты и утилиты командной строки, с помощью которых можно фильтровать сообщения, выводить их количество или заголовки, а также получить еще кучу всяких мелких удобств, о которых вы узнаете, заглянув на <http://www.gnu.org/software/mailutils/mailutils.html>. На сайте есть раздел, где разработчики приглашают обращаться с предложениями по улучшению или добавлению функций, которые пользователи сочли необходимыми — в том-то и прелесть свободного ПО, и здесь мы наблюдаем его преимущества в действии!

Пример отправки письма заданному адресу:

```
$ mail -s "текст письма" irairache@gmail.com
```

Популярный почтовый клиент *Mozilla Thunderbird* также можно эксплуатировать через интерфейс командной строки. Возьмем, например, да и отправим легким движением руки письмо со вложением (теперь вам уже должно быть ясно, как впоследствии вводить «сменную часть» с клавиатуры, через переменные):

```
ira@box:$ thunderbird --compose "to='info@linuxform.com',subject='тема письма',attachment='file:///home/ira/fotos.rar',body='текст письма'"
```

» **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Также с помощью комбинации скрипт + *Mozilla Thunderbird* можно извлекать данные из хранилища писем и вносить изменения в его структуру.

Работа с графикой

Вы удивитесь, но в окно терминала можно даже вывести настоящую картинку (см. **LXF**Hotpicks, стр. 103) — что уж говорить об автоматизации работы с изображениями. Для этого пригодится консольный графический редактор *ImageMagick*. Перечислим его самые интересные возможности:

- » наложение форм, рамок или надписей на изображение;
- » всевозможные трансформации (повороты, искривления и т. д.);
- » водяные знаки;
- » работа со сверхогромными изображениями (терапиксельными);
- » склеивание изображений друг с другом;
- » создание собственных фильтров изображений;
- » создание GIF-анимаций из группы изображений;
- » создание скриншотов;
- » выполнение скриптов на языке Magick Scripting Language.

В порядке примера использования утилиты *convert* (из состава *ImageMagick*) запишем мега-скрипт — пусть он уменьшит некое изображение **someimage.jpg** в 5 раз (т.е. сделает его размером в 20% исходного), приведет его к нужной ширине и высоте, а затем вырежет из него кусок и поместит в другой файл:

```
convert someimage.jpg -resize 20% $(basename someimage.jpg).jpg
```

```
convert someimage.jpg -resize 600x400 $(basename someimage.jpg).jpg
```

```
convert someimage.jpg -resize 72x -crop 72x72+0+0 "someimagecropped.jpg"
```

Обработка мультимедиа-файлов

Для обработки аудио- и видеофайлов используется утилита *ffmpeg*. Вот что она умеет (и не только это):

- » конвертировать мультимедиа-файлы из одного формата в другой;
- » создавать потоковые сервера для видеотрансляций (с помощью команды *ffserver*);
- » проигрывать видео и аудио (с помощью команды *ffplay*).

Извлекаем из видео в формате AVI звук и сохраняем его в формате MP3:

```
ffmpeg -i some_video.avi -vn -ar 44100 -ac 2 -ab 192 -f mp3 some_video_sound.mp3
```

Сохраним кадры из видео как отдельные изображения:

```
ffmpeg -l some_video.mpg some_video%d.jpg
```

Есть и другие утилиты для работы с видео- и аудиозаписями — *Mencoder*, *HandBrake*, *TransCode*... короче, этого добра хватает.

Обратно в GUI

Вы, видимо, уже вошли во вкус написания скриптов, а про графический интерфейс и думать забыли. Ну, а если вы хотите предоставить готовые скрипты своим менее продвинутым друзьям? Сделайте для скриптов графический интерфейс! Для этого существует утилита *Zenity*. Она позволяет использовать 15 различных видов диалоговых окон — не густо, но для подтверждения согласия пользователя или предоставления ему выбора из нескольких вариантов вполне достаточно. Для примера положим в переменную окно для выбора файла:

```
FILE=$(zenity --file-selection --title=>Выберите файл<)
```

Заключение

Какие бы задачи ни приходилось решать с помощью Linux, Shell Script поможет автоматизировать любой рабочий процесс. На нашем уроке мы получили представление об основных приемах работы со скриптами оболочки и увидели, что решаются практически все вопросы. А если вы затеваете автоматизацию в Windows-системе, обратите внимание на языки Power Shell и VBScript.

Надеемся, что если вы еще не имели опыта использования скриптов, то непременно захотели его приобрести. Как вам, например, такие варианты скриптов-помощников:

- » Скрипт для конвертации всех ODT-файлов, расположенных в конкретном каталоге, в PDF-формат.
- » Скрипт, раскладывающий все пользовательские файлы в системе в соответствии с месяцами их последнего изменения.
- » Скрипт, который запускается при загрузке и приводит рабочий стол в состояние боеготовности (открывает документы, отредактированные последними, определенные вкладки браузера и т. д.).
- » Скрипт, который записывает параметры системы пользователя в текстовый файл и отправляет на удаленный сервер.

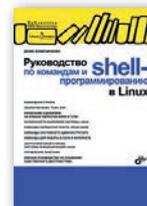
Впечатляет? Ну так дерзайте! Удачи вам. **LXF**

Литература по теме

» Ричард Блум, Кристина Бреснахэн «Командная строка Linux и сценарии оболочки»
Пожалуй, самое информативное издание на русском языке, посвященное командной строке Linux и языку Shell Script. 784 страницы справочной информации о командах для всевозможных существующих интерпретаторов (практически для всех).



» Денис Колисниченко «Руководство по командам и shell-программированию в Linux»
В книге много примеров (с подробными комментариями) готовых скриптов, написанных для интерпретаторов *bash* и *tcsh*. Описаны вопросы управления пакетами данных и работы Z Shell (*zsh*). Небольшое карманное издание о самых важных аспектах Shell Script-кодинга.



» Денис Колисниченко «Командная строка Linux и автоматизация рутинных задач»
Для читателей с высоким уровнем предварительной подготовки. В книге дана информация по узким и сложным аспектам Shell Script-программирования: файловая система *eCryptfs*, псевдофайловая система */proc*, RPM-пакеты и т. п. В издание включены подробные справочники по использованию утилит *chage*, *pwck*, *grpck*, *groupmod*, *groupdel*, *dd* и т. д.



» Марк Г. Собель «Linux. Администрирование и системное программирование»
Справочник по использованию команд оболочки для системных администраторов. Книга рассказывает об особенностях программирования в Linux-системе; детально раскрыты аспекты использования скриптов оболочки в связке с *Vim*, *Emacs*, *rsync* и *sed*. В приложении приведен полный справочник команд оболочки.



FreeCAD: СВОЙ корпус для Pi

Робин де Йонг демонстрирует, как создать корпус для Raspberry Pi и украсить его своим текстом или логотипом, применив Python и *FreeCAD*.



Наш эксперт

Робин де Йонг много лет проработал инженером-проектировщиком, но предпочитает учить других, и делится своими изобретениями в традициях открытого кода.

Если вы хоть раз вскрывали упаковку нового Apple или Blackberry, то испытаете сильное разочарование, когда вам доставят ваш первый Raspberry, потому что сходство будет чисто плодоягодным [Apple — яблоко, Blackberry — ежевика, Raspberry — малина, — прим. пер.]. Pi прибывает в простой белой картонной коробке, обернутой антистатическим покрытием, с единственным печатным листком с нормативной информацией.

Самое примечательное в Raspberry Pi то, что это просто плата, в точности как материнская плата ПК, и выглядит она довольно голый. Поставьте Pi на стол — и стол будет поцарапан. Сядьте на него — и в вас волеются контакты. Оставьте его без присмотра — и на него наступят, или его растерзает собака, или начнет грызть ребенок. Для Pi остро необходим корпус; им мы и займемся на данном уроке.

С этой целью установите *FreeCAD* на свою машину Linux, а чтобы корпус распечатать, вам нужен или 3D-принтер, или друг с 3D-принтером, или несколько фунтов, чтобы использовать онлайн-сервис 3D-печати типа i.materialise.com или shapeways.com.

Для придания индивидуальности вашему корпусу Pi понадобится также установить *Inkscape*, но этот шаг опциональный. Весь код и прочие файлы для нашего руководства имеются на **LXF DVD** этого месяца и на Github — <https://github.com/dejongh/raspberrypicase>.

Возьмите свой Raspberry Pi и проверьте, какая это модель — А или В. Модель В определяется по двум разъемам USB. В данном руководстве мы предполагаем, что у вас Модель В, но вы легко можете изменить часть размеров в своем файле CAD, чтобы сделать корпус для Модели А. Сначала надо измерить некоторые важные части Pi. Найдите на плате монтажные отверстия. Их должно быть два (см. схему внизу страницы).

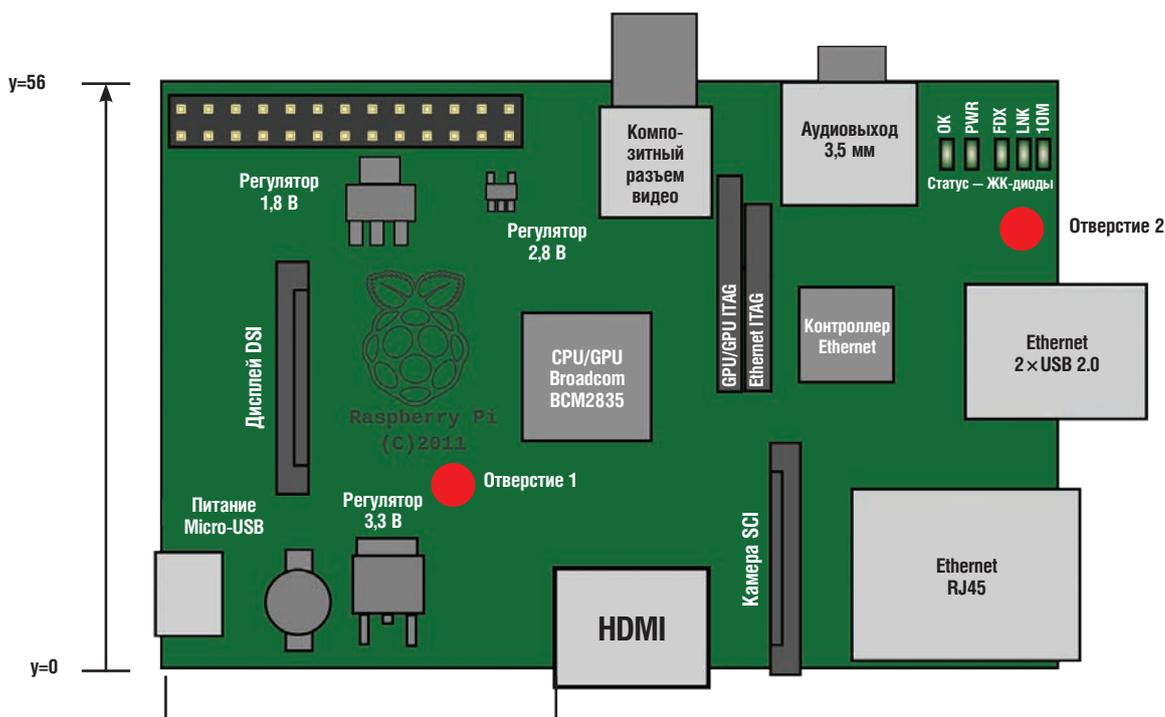
FreeCAD для программ на Python

Открыв *FreeCAD*, вы увидите экран приветствия. Выберите опцию **Part Design** [Конструирование части]. Чтобы настроить *FreeCAD* на код Python, перейдите в **View > Views** [Вид > Вид] и отметьте **Python Console**. Перейдите в **Edit > Preferences > General > Output window** [Править > Параметры > Общие > Окно вывода] и отметьте две опции **Redirect** [Перенаправить] внизу экрана, затем щелкните по **Apply** [Применить] и **OK**. Теперь перейдите во **View > Views** и отметьте **Report view**.

Начнем с импорта требуемых библиотек. Введите следующее в консоли Python (аккуратнее с регистром букв!):

```
import Part
from FreeCAD import Vector
```

Теперь мы сможем применить модуль **Part** для создания таких общих 3D-форм, как кубоиды, цилиндры и сферы, и модуль **Vector**,



» На плате два монтажных отверстия. Мы настроим модель CAD на $x=0$ и $y=0$ в нижней левой точке платы Pi, чтобы было проще делать замеры.

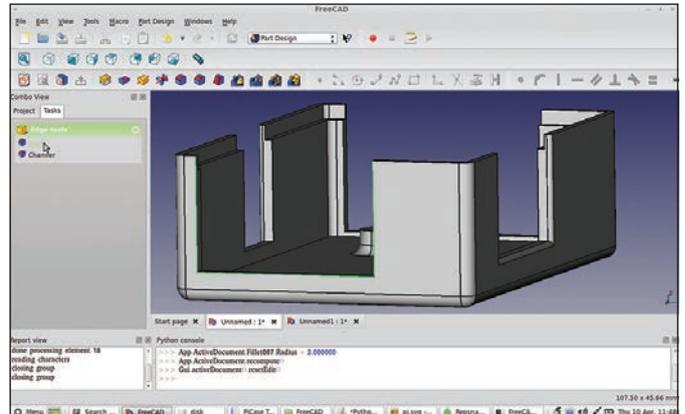
Украшим вашу модель скруглениями

Взглянем правде в глаза — наша модель CAD на данной стадии разработки выглядит ужасно, но такого и следовало ожидать. К счастью, есть один трюк, благодаря которому мы добьемся, чтобы наши объекты выглядели профессионально — и он именуется функция fillet [скругление].

Выделение любого края и нажатие на кнопку Fillet мгновенно сделает его прекрасным. Удерживайте Ctrl и выберите более одного края, и мы можем получить скругления, которые идут вокруг основы и стенок корпуса Pi. Попробуйте сделать это прямо сейчас.

Вам нужно будет использовать 2,5 мм для дна и внешних уголков. Для внутренних уголков используйте 1,5 мм. Когда вы закончите, модель сразу обретет более шикарный и профессиональный вид. Любые скругления можно изменить или удалить, так что не бойтесь пробовать.

► Радиус любого скругления можно настраивать на ходу. Скругления прямо-таки преображают визуальную привлекательность корпуса.



чтобы размещать эти формы. Теперь взгляните на свой Raspberry Pi: представим себе, что угол возле разъема питания — это точка начала координатных осей x и y, а нижняя сторона платы — это ноль на оси z. Это облегчит нам задачу ввода векторов, соответствующих замерам, которые мы будем делать на плате Raspberry Pi. Возьмите линейку или штангенциркуль и измерьте расстояние от нижнего левого угла платы до разъема питания. Измерьте и запишите показатели x и y для каждого отверстия.

Создание основной коробки

Начнем с создания базового параллелепипеда:

```
boxshape=Part.makeBox(93,64,30,Vector(-4,-4,-8))
```

Мы создали параллелепипед, пометив его boxshape, с длиной 93 мм, шириной 64 мм и высотой 30 мм, и переместили его так, чтобы угол платы оставался в точке 0; это позволит нам ввести любые замеры платы как векторы, не прибегая к математическим расчетам, от которых голова пухнет.

```
cutout=Part.makeBox(87,58,30,Vector(-1,-1,-5))
```

А здесь мы сделали еще один параллелепипед, назвав его cutout, всего на 3 мм меньше, чтобы потом вырезать его из первого, таким образом:

```
pibase=boxshape.cut(cutout)
```

Вы заметите, что ничего не произошло, но дело в том, что мы должны велеть FreeCAD нарисовать результат. Сделайте это, набрав

```
Part.show(pibase)
```

Щелкните правой кнопкой по основному окну и выберите Navigation styles [Стили навигации] > Blender. Затем увеличьте масштаб средней кнопкой мыши, и удерживайте, чтобы вывести вид. Теперь вы видите правильно отображенный результат.

Это будет основа нашего корпуса Pi, и нам надо, чтобы корпус был на 4 мм больше самой платы по всем показателям. Нам нужны входы для всяких штекеров, которые мы будем подсоединять к гнездам на Raspberry Pi, так что некоторые области на корпусе мы прорежем. Нужно просто отмерять от ранее определенного нами нулевого угла Pi и вводить эти данные как векторы. Попробуем!

```
leftsidecut=Part.makeBox(20,42,30,Vector(-10,2,-4))
```

```
pibase=pibase.cut(leftsidecut)
```

Это позволило вырезать отверстие шириной 42 мм, чтобы вставлять в него штекер питания и SD-карту. Давайте продолжим,

сделав то же самое на всех четырех сторонах (ваши замеры могут отличаться).

```
hdmicut=Part.makeBox(18,20,30,Vector(35,-10,-2))
```

```
avcut=Part.makeBox(33,20,30,Vector(38,50,0))
```

```
usbcut=Part.makeBox(20,38,30,Vector(80,1,-2))
```

```
pibase=pibase.cut(hdmicut)
```

```
pibase=pibase.cut(avcut)
```

```
pibase=pibase.cut(usbcut)
```

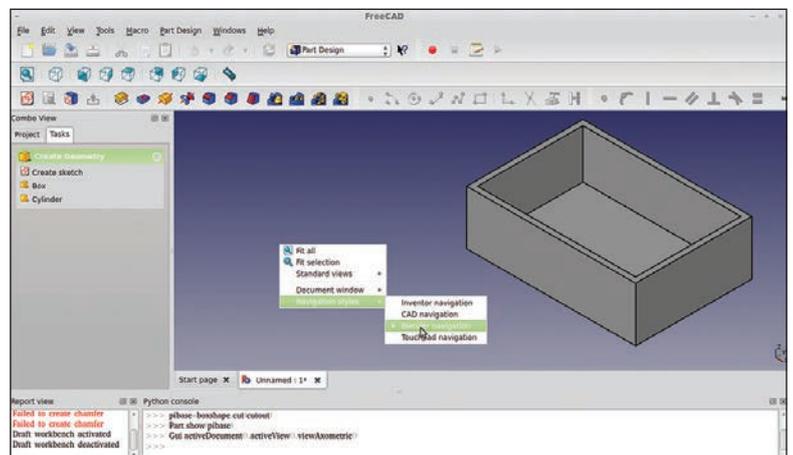
Теперь посмотрим, что получилось, повторив команду показа: Part.show(pibase)

В этот момент нам, возможно, придется спрятать более ранние геометрические фигуры, поэтому перейдем во вкладку Project, выберем Shape, перейдем во вкладку View и изменим Visibility на False. Нам понадобится еще и крышка для корпуса, а также место, чтобы ее пристроить. Давайте вычтем форму крышки из стенок PiBase:

```
lidcut=Part.makeBox(90,61,10,Vector(-2.5,-2.5,19))
```

```
pibase=pibase.cut(lidcut)
```

И последнее, что нам нужно на корпусе — штырки, чтоб садить на них плату Pi, закрепив ее. Мы можем сделать это с помощью команды Part.makeCylinder. Помните, что мы создаем модель, которая начинается с нижней стороны платы, поэтому



► Интерфейс FreeCAD можно настроить на запуск скриптов Python из консоли внизу. Настройте простую навигацию мышью, выбрав стиль Blender.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

должно быть легко создать цилиндр, который закончится в $z=0$. Подобно `makeBox`, `makeCylinder` требует определенных величин, определяющих размер цилиндра, в форме (радиус, высота). Кроме того, легко найти центр отверстий, потому что мы можем просто отмерять от края платы и использовать векторы.

```
boss1=Part.makeCylinder(3.5,7,Vector(25.5,18,-7))
```

Обратите внимание, что мы ввели ту же самую цифру для высоты и для вектора оси z , но присвоили вектору z минус. Благодаря этому возвышение начнется на нижней стороне платы Raspberry Pi. Повторите то же самое для второго штырька:

```
boss2=Part.makeCylinder(3.5,7,Vector(80,43.5,-7))
```

Пора присоединить штырьки к основе. Здесь мы воспользуемся командой `fuse`, которая работает точно так же, как и команда `cut`:

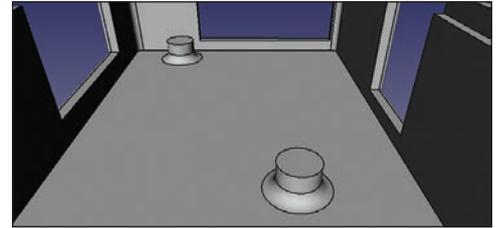
```
pibase=pibase.fuse(boss1)
```

```
pibase=pibase.fuse(boss2)
```

И вот вам законченный корпус; только выглядит он так себе. Чтобы исправить эти острые углы и края, следуйте инструкциям во врезке на стр. 71, «Украшим вашу модель скруглениями», и вы получите более профессиональный вид! **LXF**

Фиксируем Pi в корпусе

Мы создали штырьки для платы Pi, чтобы насадить ее на них, но не сделали в них отверстий. Почему? Домашние и потребительские 3D-принтеры не имеют достаточного разрешения печати, чтобы обеспечить создание отверстий, и, кроме того, нельзя быть на 100% уверены, что отверстия совпадут с отверстиями в плате Pi. Лучше будет обойтись дрелью. Разместите Pi внутри корпуса и наметьте центр отверстий кончиком

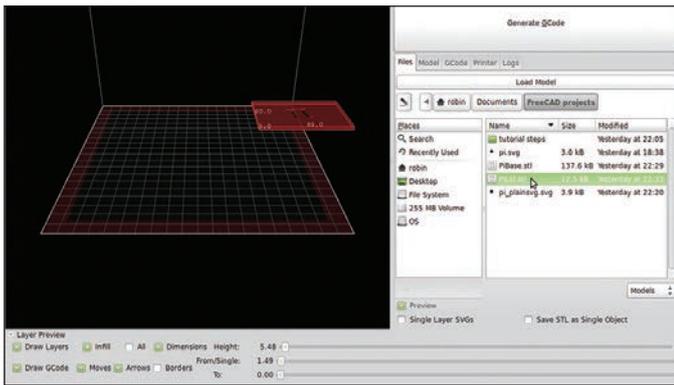


➤ Просверлите небольшое отверстие и прикрутите Pi к корпусу, чтобы он не дергался при вставке или отсоединении штекеров.

сверла. Теперь вытащите Pi и проделайте отверстия сверлом, которое немного тоньше шурупа.

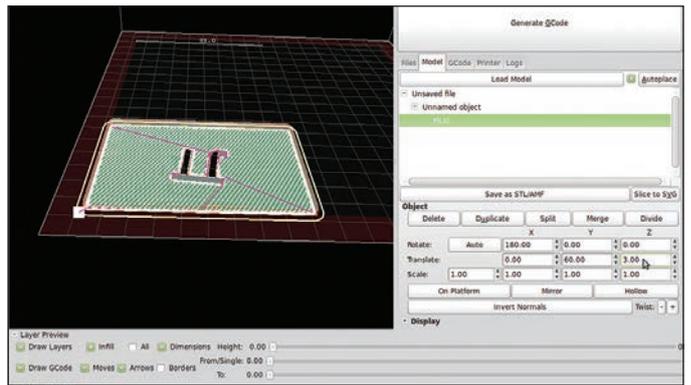
Тогда, если вы вставите шуруп и ввернете его, он сам проложит себе путь в пластик.

3D-печать корпуса Pi



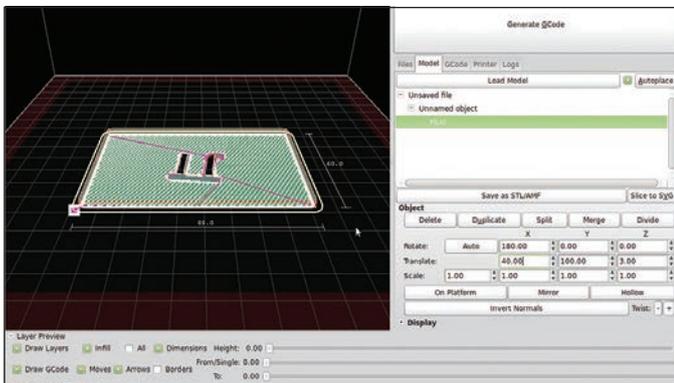
1 Импортируем файл STL

Ваш корпус и крышка должны быть экспортированы как файлы `.stl`, и теперь вы можете импортировать их по одному за один раз и печатать. Перейдите в свою программу печати и выберите файл `PiLid.stl`. Щелкните по `Load model` [Загрузить модель], и вы должны увидеть, как он появляется в основном окне.



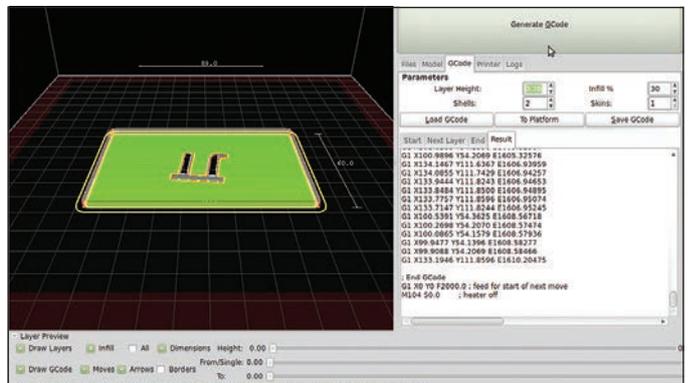
2 Ориентируем печать

Чтобы сделать симпатичную отделку по верху крышки, поместите верх на платформу печати и используйте настройку нагрева платформы. Это означает, что сначала надо перевернуть печать вверх ногами. Введите 180 в окно x -rotation и настройте z translation на 3.



3 Центруем объект

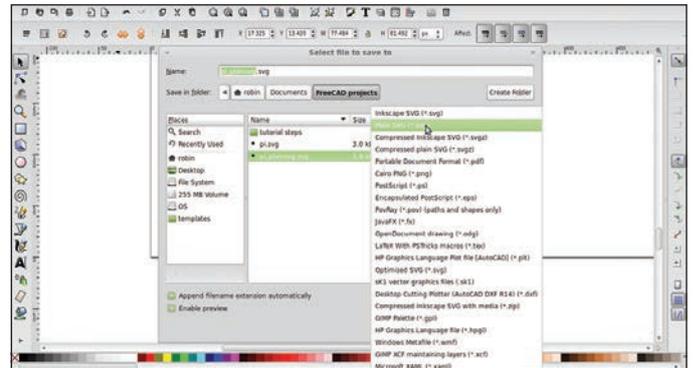
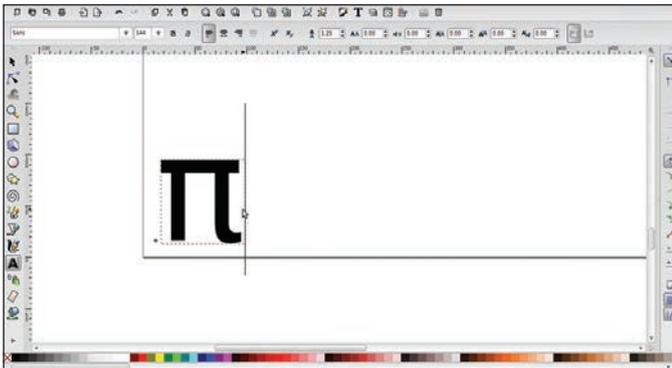
Убедитесь, что ваш текст (см. «Придадим индивидуальности корпусу») выглядит задом наперед. Настройте смещение x и y , чтобы разместить ваш объект на платформе печати. Введите параметры печати в соответствии с рекомендациями вашего принтера. Нажмите `Generate G-code`, чтобы нарезать получившуюся у вас модель на слои.



4 Запускаем 3D-принтер

Убедитесь, что используется опция нагрева платформы, чтобы избежать деформации и обеспечить хорошее прилегание к платформе печати. Если вы используете стеклянную платформу (рекомендуется), у вас заодно получится отличная отделка. Когда крышка напечатается, повторите то же самое с PiBase, которая должна печататься обычным способом, снизу вверх.

Придаем индивидуальности корпусу

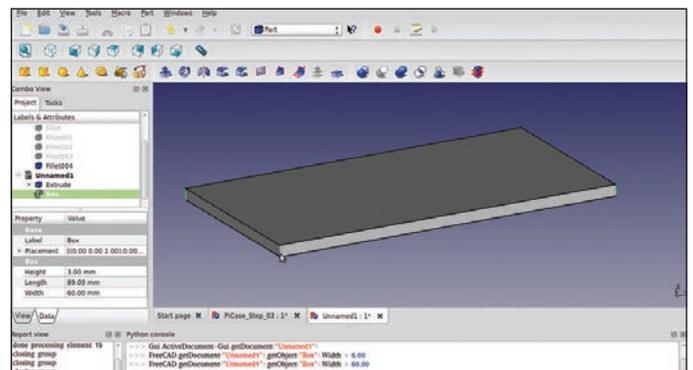
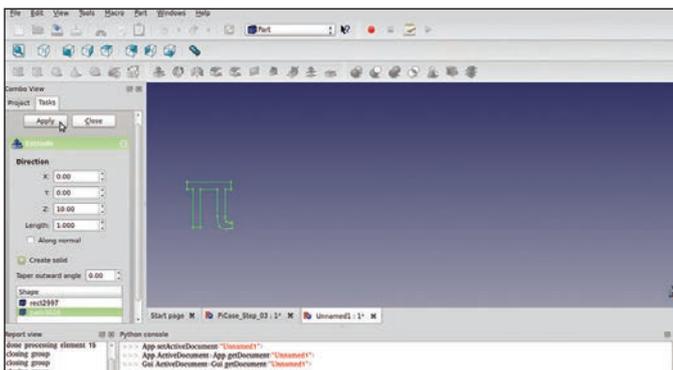


1 Создадим текст в Inkscape

Откройте *Inkscape* и выберите текстовую кнопку на левой панели инструментов. Щелкните по любому месту на странице и введите текст, который хотите разместить на крышке корпуса. Я выбрал символ «пи», так что мне нужно будет скопировать его из утилиты спецсимволов (character map), предустановленной в Linux.

2 Экспортируем текст из Inkscape

Нажмите F1, затем выделите текст. Перейдите в Path > Object to path [Контур > Преобразование объекта в контур]. Перейдите в File > Save As [Файл > Сохранить как] и выберите Plain SVG (.svg) из выпадающего списка. Сохраните файл. Во FreeCAD перейдите в File > Import и выберите только что сохраненный файл. Нажмите Open, а затем в качестве Geometry [Геометрия] выберите SVG.

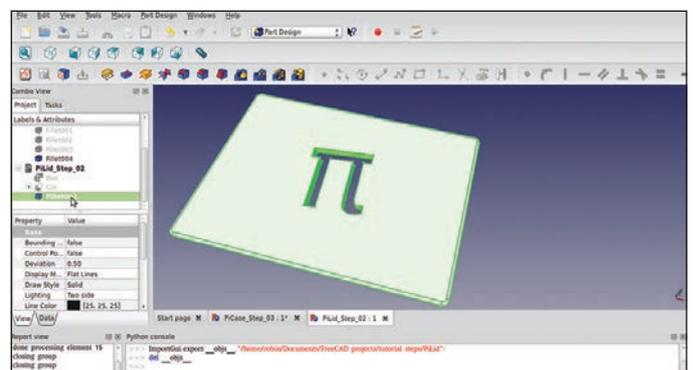
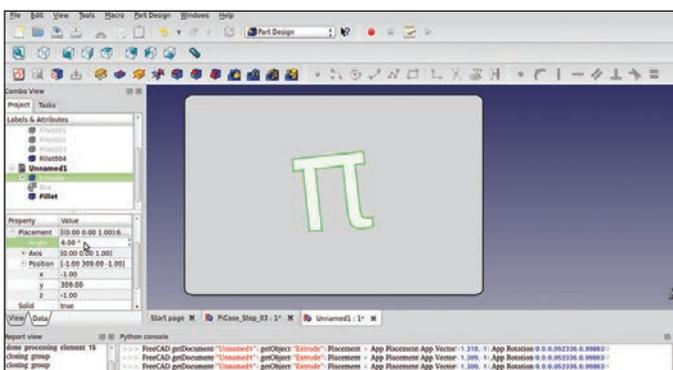


3 Импортируем SVG-геометрию во FreeCAD

Ваша SVG-геометрия импортируется как сгруппированный объект с именем path... Теперь выберите его из списка проектов. Он должен позеленеть, чтобы вы удостоверились в его выделении. Убедитесь, что вы находитесь в инструменте Part [Часть] (выпадающий список в центре наверху), затем выберите кнопку Extrude a selected sketch [Сформировать выбранный эскиз]. Введите 10 для z: и выберите Create solid [Создать сплошной], после чего нажмите Apply [Применить].

4 Создадим индивидуальную крышку

Нажмите на желтую кнопку Create a box solid [Создать сплошной параллелепипед], затем измените размер, как показано в окнах Height, Length и Width [Высота, Длина и Ширина]. Выберите все четыре угловых края, удерживая Ctrl, и примените скругление 2 мм (вы должны быть в инструменте дизайна Part). Теперь вернитесь в Project list [Список проектов] и выберите Extrude group [Сформировать группу].



5 Вырежем буквы SVG

Щелкните по вкладке Data [Данные] и выберите Placement [Размещение]. Перейдите в Position [Положение] и измените параметр z на -1. Теперь настраивайте параметр u, пока ваши буквы не попадут в должное положение на крышке. Настройте параметр Angle [Угол], если хотите, чтобы текст был размещен под углом. Настройте параметр x, чтобы добиться желаемого расположения.

6 Экспортируем в 3D-принтер

Удерживая Ctrl, выберите First Fillet then Extrude [Скруглить, затем сформировать] и щелкните по кнопке Cut [Нарезать]. Выберите крышку из списка проектов и проверьте, чтобы все стало зеленым. Перейдите в File > Export и выберите Mesh formats [Форматы сетки]. Введите filename.stl. Повторите то же с корпусом Pi. Теперь вы готовы к 3D-печати!

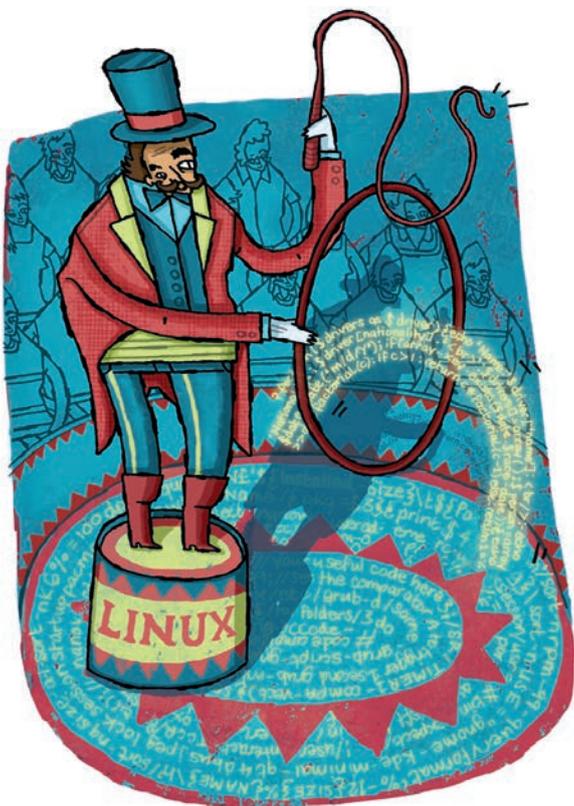
DevStack. Как создать облачко

Джолион Браун показывает, как с помощью *DevStack* — «упертого скрипта» — развернуть облако *OpenStack* в одной виртуальной машине.



Наш эксперт

Свободными вечерами **Джолион Браун** автоматизирует построение инфраструктуры, чтобы его дети могли создавать сервера *Minecraft* сами, не приставая с этим к нему.



Скорая помощь

Сейчас стоит сделать снимок новой виртуальной машины. В менеджере *VirtualBox* нажмите *Snapshots* [Снимки] вверху справа, затем нажмите на иконку *Take Snapshot* [Сделать снимок] (маленький голубой фотоаппарат). Это избавит вас от полной переустановки, если что-то пойдет не так.

Последний раз мы серьезно занимались *OpenStack* еще в **LXF161** («Что такое *OpenStack*», стр. 54). С тех пор проект существенно повзрослел и вырос, придерживаясь своего полугодового цикла выпуска релизов (девятый релиз с кодовым названием *Icehouse* вышел в середине апреля 2014 года) и получив поддержку крупных компаний, таких как IBM и Red Hat. Действительно, сообщество *OpenStack* растет с огромной скоростью, поддерживаемое независимым фондом с около 16000 зарегистрированных участников.

Этот рост популярности *OpenStack* происходит на фоне ускоряющегося роста облачных вычислений во всевозможных видах как платформы для бизнеса и корпораций. Раньше компании могли только приобретать или размещать физические серверы в дата-центрах. На этом рынке есть чем заняться, и *OpenStack* с его подачей как «открытой облачной операционной системы», кажется, набирает темп.

То, что освоить *OpenStack* стоит всем приверженцам Linux, видно с первого взгляда; но с чего начать? Выбор у тех, кто интересуется подобными системами, впечатляющий: публичные

облака от Rackspace и HP, версии, адаптированные под продукты, типа *Piston OpenStack*, да еще и пакеты, имеющиеся в большинстве крупных дистрибутивов Linux (более подробный список различных стартовых вариантов см. на www.openstack.org/software/start). Также стоит зайти на <http://docs.openstack.org> и попробовать одно из руководств для конкретных дистрибутивов. Но в этой статье мы поговорим об альтернативе: *DevStack*.

Про DevStack

DevStack называет себя «упертым скриптом» (по сути, это скрипт оболочки); он прекрасно подходит для сравнительно быстрой настройки и запуска *OpenStack* в локальной среде. На нашем уроке мы будем запускать его в виртуальной машине (ВМ) с помощью *VirtualBox*, но еще лучше воспользоваться обычным компьютером (на котором к тому же все будет работать быстрее). Наши инструкции подходят к каждому из этих вариантов.

Учтите, просто запустить установочный скрипт в вашем настольном дистрибутиве вряд ли получится: *DevStack* автоматически загружает и настраивает кучу программ. Причем все это не только, скорее всего, не заработает сразу, но и потребует вечности на очистку.

Также стоит отметить, что *DevStack* определенно не подходит для работы в промышленных масштабах (так, например, у вас не получится создать с его помощью конкурента для облака Amazon). Как поясняется в FAQ на сайте <http://devstack.org>: «Некоторые особенности реализации *DevStack* делают его непригодным для реальных сред. Вы предупреждены!» Не последние по значимости из этих особенностей касаются безопасности: пароли хранятся в файлах, а настройки *DevStack* полностью открыты через **sudo** для пользователя, который выполняет установку.

Для целей нашего урока предположим, что установка выполняется в типовой домашней сети (хотя, может, и не слишком типичной для читателей *Linux Format!*) с интернет-роутером/ADSL-модемом, который выступает как шлюз, перенаправляет DNS-запросы и служит DHCP-сервером для остальных устройств сети. В этих примерах устройство имеет IP-адрес 192.168.0.1, а маска подсети — 255.255.255.0 (т.е. это сеть /24). Виртуальные машины, которые мы создадим в своем облаке, будут доступны с компьютера Linux, на котором запущен *VirtualBox*, и ниоткуда больше.

В качестве отправной точки для установки *DevStack* лучше использовать минимальный образ дистрибутива. Мы воспользуемся ISO-образом Ubuntu 12.04 LTS, который есть на **LXF DVD** (<http://bit.ly/1eQePdF>) с объемом около 30 МБ (альтернативы — Fedora и CentOS/RHEL). Выберите образ в соответствии со своим чипсетом (32- или 64-битным — мы пользуемся вторым). Запустите *VirtualBox* и выберите *Preferences > Network* [Настройки

> Сеть]. Создайте две сети только с хостами: vboxnet0 (с IP-адресом 172.16.0.254 и сетевой маской 255.255.0.0) и vboxnet1 (10.0.0.1/255.0.0.0). Теперь создайте новую VM, выделив ей от 2 ГБ оперативной памяти (чем больше, тем лучше). К сожалению, во время написания данной статьи мы столкнулись с ошибкой в QEMU, из-за которой пришлось отключить опцию Enable I/O APIC [Включить ввод/вывод через APIC]. А это, кажется, привело к тому, что *VirtualBox* смог использовать только один процессор для гостевой системы. У вас все может быть иначе!

Создаем новый хост

Размер системного диска VM мы задали равным 25 ГБ. Для 64-битной VM мы также выбрали опцию Enable PAE/NX [Включить PAE/NX]. В настройках сети VM включите адаптер 1 [Adaptor 1] как «мостовой адаптер [Bridged Adaptor]» с названием eth0. Адаптер 2 [Adaptor 2] нужно включить как «адаптер хоста [Host-only Adaptor]» с названием vboxnet0. Адаптер 3 [Adaptor 3] аналогичен второму, но имеет имя vboxnet1. Остальные настройки можно оставить без изменений. Вставьте образ *mini.iso*, содержащий образ Ubuntu, в виртуальный DVD-привод и запустите виртуальную машину. (Параметры, примененные на нашем уроке, см. на рис. 1.)

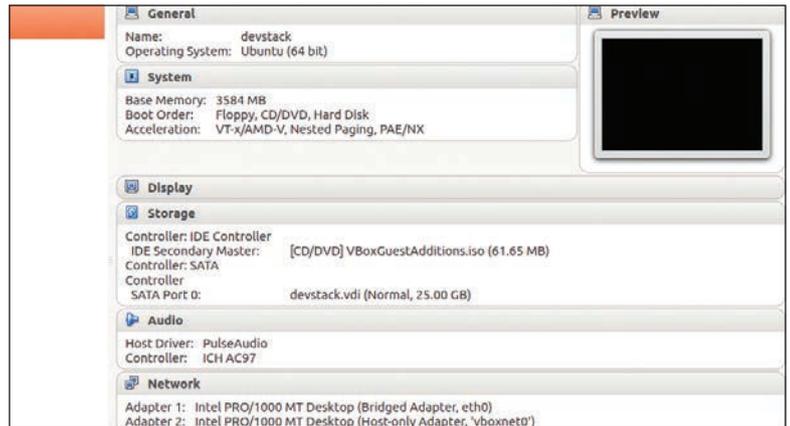
Начнется установка в текстовом режиме. Большинство параметров не требуют пояснений и знакомы всем, кто когда-либо устанавливал Linux (а к ним, видимо, относится большинство читателей этого журнала). Задайте имя хоста *devstack* и для простоты разрешите установщику использовать весь диск. Когда вам предложат создать пользователя, создайте одного пользователя *stack* с любым сходным паролем. При выборе пакетов возьмите базовые опции Ubuntu и OpenSSH (заметьте, что доступна и опция *OpenStack* — но ее выбирать не надо). Через некоторое время установка завершится, VM перезагрузится (не забудьте извлечь DVD из виртуального привода, когда установщик предложит это сделать), и появится строка входа в систему. Введите логин *stack* и созданный пароль. Затем не помешает выполнить быстрое обновление пакетов (надеюсь, их будет не слишком много):

```
sudo apt-get update && sudo apt-get upgrade
```

После этого нужно добавить пакет *Git* (со всеми его зависимостями), который будет интенсивно использоваться во время установки *DevStack*:

```
sudo apt-get install git -y
```

Подлинно хорошая идея — задать для хоста статический IP-адрес. Для этого отредактируйте файл `/etc/network/interfaces`:



```
sudo vi /etc/network/interfaces
```

Измените запись для eth0 следующим образом (здесь мы используем адрес 192.168.0.100; вместо него укажите свой):

```
auto eth0
iface eth0 inet static
address 192.168.0.100 #your IP
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1 #your router/gateway IP
dns-nameservers 192.168.0.1
```

Также нужно добавить в этот же файл следующие строки:

```
# OpenStack Public Interface
auto eth1
iface eth1 inet static
address 172.16.0.1 #the virtual DevStack created IP
netmask 255.255.0.0
broadcast 172.16.255.255
network 172.16.255.255
# OpenStack Private Interface
auto eth2
iface eth2 inet manual
up ifconfig eth2 up
```

Внеся изменения, перезапустите интерфейс eth0 и убедитесь, что с параметрами все нормально, с помощью *ifconfig*:

```
sudo ifdown eth0 && sudo ifup eth0 ifconfig eth0
```

► Рис. 1. После небольшой настройки в *VirtualBox* можно запустить наше облако *OpenStack*.

Что такое OpenStack?

Изначально разрабатываемый Rackspace и НАСА (к ним быстро присоединились и другие компании), проект *OpenStack* был запущен в 2010 году, чтобы позволить всем запускать облачные сервисы на бытовом «железе». В 2012 году создали независимую контролирующую организацию проекта, и он стал описываться как «облачная ОС, способная управлять большими пулами вычислительных и сетевых ресурсов и хранилищ дата-центра с помощью панели управления, которая передает управление администраторам и предоставляет их пользователям ресурсы через web-интерфейс». Эта ОС состоит из множества компонентов, написанных разными программистами под флагом *OpenStack*.

► **OpenStack Compute** (кодировое название *Nova*) позволяет инициализировать VM и управлять ими.

► **OpenStack Networking** (кодировое название *Neutron*) — подключаемая система API для управления сетями и IP-адресами.

► **OpenStack Object Storage** (кодировое название *Swift*) — сервис хранения объектов.

► **OpenStack Block Storage** (кодировое название *Cinder*) — сервис хранения блоков данных.

► **OpenStack Identity** (кодировое название *Keystone*) — общая система аутентификации для облака *OpenStack*.

► **OpenStack Image Service** (кодировое название *Glance*) — сервисы обнаружения, регистрации и загрузки образов дисков и серверов.

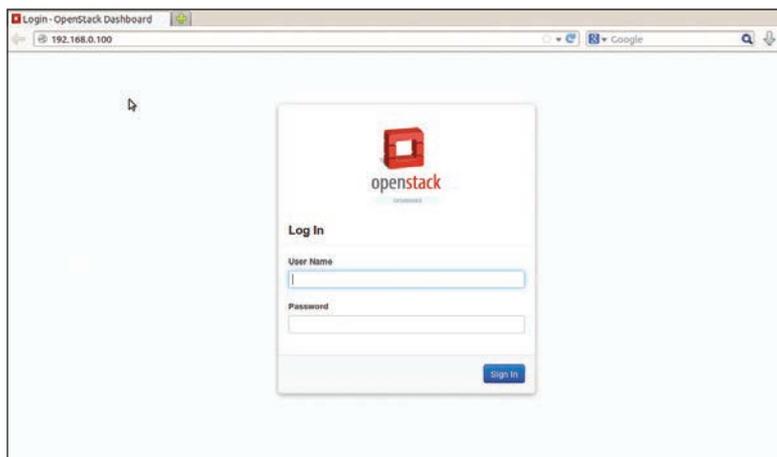
► **OpenStack Dashboard** (кодировое название *Horizon*) — графический интерфейс для доступа, инициализации и автоматизации ресурсов.

► **OpenStack Telemetry** (кодировое название *Ceilometer*) поддерживает централизованный сбор данных измерений и мониторинга.

► **OpenStack Orchestration** (кодировое название *Heat*) позволяет выполнять аранжировку на основе шаблонов.

На различных стадиях утверждения в качестве официальных программ *OpenStack* находится немало других проектов: сервис баз данных (*Trove*), сервис предоставления «голого железа» (*Ironic*), сервис очередей (*Marconi*) и сервис обработки данных (*Sahara*). Цель сервиса *Sahara* — в удобном предоставлении кластеров *Hadoop* поверх *OpenStack*. Все эти программы доступны по лицензии Apache 2.0 (и, разумеется, работают в Linux).

► Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



➤ Рис. 2. Наконец-то! Появилось окно входа в мифическую панель управления OpenStack.

Теперь интерфейс должен использовать указанный вами адрес. Запустите интерфейсы eth1 и eth2:

```
ifup eth1
ifup eth2
```

Наконец, нужно дать пользователю stack права на запуск команд с `sudo` без пароля. В обычных условиях подобная идея даже не плоха, а **ОЧЕНЬ** плоха, но здесь такое сделать необходимо, иначе *DevStack* не установишь:

```
sudo -i
echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
exit
```

Установка DevStack

Проект *DevStack* часто обновляется (как и все проекты, имеющие отношение к *OpenStack*, особенно во время выхода большого релиза). К счастью для нас, довольно легко извещать, что нам нужна определенная версия *DevStack*, указав соответствующую ей ветку командой `git clone` (на момент написания статьи наиболее стабильным был релиз Havana):

```
git clone https://github.com/openstack-dev/devstack.git -b stable/havana devstack/
cd devstack
```

После загрузки копии скриптов *DevStack* нужно создать файл `localrc` для указания потребных нам сервисов *OpenStack*. Поскольку наш урок в общем начальный, мы воспользуемся *Nova*, *Cinder*, *Glance*, *Horizon* и *Keystone* (во врезке «Что такое *OpenStack*?» на стр. 75 можно узнать, что предоставляет каждый из этих сервисов).

Во избежание лишних сложностей мы ограничимся встроенным сетевым компонентом *Nova* взамен гораздо более мощного *Neutron*. Шаблонная версия файла `localrc`, которую мы можем менять, находится в подкаталоге `samples`:

```
cp samples/localrc .
vi localrc
```

В разделе `Minimal Contents` добавьте или измените следующие параметры:

```
ADMIN_PASSWORD=devstack
MYSQL_PASSWORD=devstack
RABBIT_PASSWORD=devstack
SERVICE_PASSWORD=devstack
SERVICE_TOKEN=token
```

По добавлении этих параметров установочный скрипт не будет запрашивать их во время установки. Обратите внимание, что все наши пароли одинаковы. Это сделано только для простоты, и в любой другой среде было бы ужасной идеей. Пониже этого

раздела (точное место не имеет значения) добавьте следующие строки для указания наших сервисов:

```
# Branches
NOVA_BRANCH=stable/havana
CINDER_BRANCH=stable/havana
GLANCE_BRANCH=stable/havana
HORIZON_BRANCH=stable/havana
KEYSTONE_BRANCH=stable/havana
```

Наконец, добавьте строки, приведенные ниже. Диапазон адресов `FLOATING_RANGE` не должен использоваться в локальной сети. Этот параметр задает диапазон IP-адресов /24, используемых как плавающие адреса для создаваемых экземпляров ВМ *OpenStack*. `FLAT_INTERFACE` — это интерфейс Ethernet, посредством которого наш хост подключается к локальной сети. Параметры `FIXED` относятся к приватной внутренней VLAN *OpenStack*.

```
FLOATING_RANGE=172.16.1.0/24
FLAT_INTERFACE=eth0
LOGFILE=/home/stack/stack.log
FIXED_RANGE=10.11.12.0/24
FIXED_NETWORK_SIZE=256
```

Сохраните файл и вернитесь в командную строку. Теперь можно запустить установочный скрипт:

```
./stack.sh
```

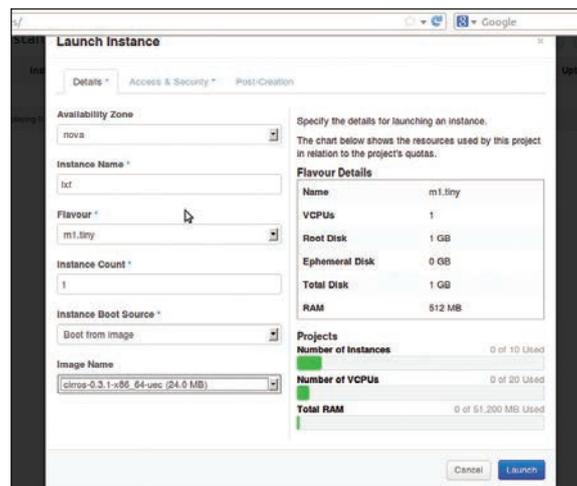
На выполнение скрипта понадобится время — успеете заварить себе чайку; а на экране по мере установки всяких пакетов и проверки кода будет пролетать череда сообщений. Допив чашечку, загляните на <http://devstack.org/stack.sh.html>, чтобы получить представление о деятельности скрипта. В конце концов вы будете вознаграждены сообщением вроде `'stack.sh completed in X seconds [скрипт stack.sh завершился за X секунд]`. В моем случае секунд было 2135. В данный момент определенно стоит сделать еще один снимок ВМ (см. «Скорая помощь», стр. 74).

Пробуем DevStack

Теперь в браузере из любой другой точки сети можно подключиться к <http://192.168.0.100> (при этом появится окно, показанное на рис. 2). В *DevStack* есть две учетных записи пользователей: `admin` и `demo`. У обоих — пароль `devstack`, благодаря ранее заданным нами настройкам. Зайдите как пользователь `admin` и перейдите на вкладку `Images` [Образы] с левой стороны окна. Вы должны увидеть здесь три образа *Cirros*. *Cirros* — это маленький дистрибутив Linux, используемый для тестирования облачных сервисов

Скорая помощь

DevStack можно остановить скриптом `unstack.sh`, расположенным в том же каталоге, что и `stack.sh`, который мы запускали для установки облака *OpenStack*. При запуске `stack.sh` происходит повторная сборка облака с нуля. Чтобы начать использовать существующие данные, запустите `rejoin-stack.sh`.



➤ Рис. 3. Cirros — готовый образ для проверки загрузки и базовых функций OpenStack.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

и поэтому прекрасно подходящий для наших целей. При желании можно добавить и другие образы. Облачные образы для Ubuntu, которые полезно попробовать, доступны на <http://cloud-images.ubuntu.com>. Выберите создание образа в меню и укажите URL-адрес образа. Отметьте образ как публичный и оставьте пустыми все параметры диска и памяти. Форматом образа должен быть QCOW2. Статус образа изменится на Queued [В очереди], а затем — при загрузке образа из Интернета и его обработке сервисом Glance — на Saving [Сохранение]. Через некоторое время (зависящее от скорости интернет-соединения) статус образа изменится на Active [Активный].

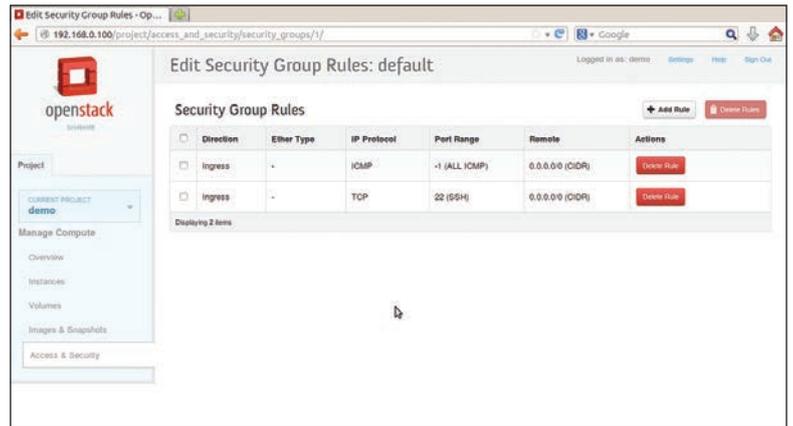
Выйдите из системы и снова зайдите как пользователь demo. Убедитесь, что текущий проект связан с пользователем demo, при необходимости изменив настройку через выпадающее меню. Перейдя на вкладку Instances [Экземпляры VM], нажмите кнопку Launch Instance [Запустить VM]. Введите параметры для запуска Cirros (как на рис. 3), при необходимости используя выпадающее меню. В частности, обратите внимание на меню Flavour и уделите внимание опциям. Для нашей системы разумной выгладит m1.tiny, но для настоящей системы на настоящем «железе» больше подойдут варианты small, medium, large и xlarge. Эти варианты можно добавлять аналогично созданию новых образов. Справа вы увидите корневой [Root] и временные [Ephemeral] диски. Корневой диск — это диск с операционной системой. Временные диски можно добавлять в виртуальную машину, но при ее остановке они исчезают.

Выдача доступа

На вкладке Access & Security [Доступ и безопасность] можно настроить пару ключей ssh и внедрить ее в загружающуюся VM. В данной статье мы это опустим. На вкладке Post-Creation [Действия после создания] можно перечислить команды для запуска после создания виртуальной машины; при необходимости это могут быть команды установки каких-либо пакетов или агентов. Нажмите кнопку Launch [Запуск], и начнется сборка нашей новой системы. В зависимости от мощности компьютера это может занять несколько минут.

Вы увидите, что на нашем новом экземпляре VM — lxf — запущена Ubuntu Precise, и ей назначен IP-адрес 10.11.12.2. Этот адрес взят из внутренней сети VLAN, которую мы настроили ранее. Чтобы VM стала доступной с нашего компьютера Linux, нужно назначить ей плавающий IP. В разделе Actions [Действия] справа нажмите на выпадающее меню и выберите Associate Floating IP [Назначить плавающий IP]. Через некоторое время рядом с исходным появится новый IP-адрес — скорее всего, он будет 172.16.1.1.

Теперь выберите вкладку Access and Security [Доступ и безопасность] в меню слева. Откроется список групп безопасности [Security Groups]; по умолчанию используется единственная, которая есть в списке. Выберите Edit Rules [Редактировать правила] справа и Add Rule [Добавить правило] в появившемся окне.



► Рис. 4. Задайте правила безопасности, чтобы пинговать хосты извне вашего облака.

Откроется окно с одноименным заголовком. В выпадающем меню Rule [Правило] выберите ALL ICMP, оставьте все остальные поля неизменными, затем нажмите Add [Добавить]. Это позволит нам пинговать хосты извне нашего облака. Добавьте еще одно правило, на сей раз выбрав SSH. (Экранный снимок см. на рис. 4.) Эти правила применяются сразу, поэтому вернитесь в наш компьютер с Linux и откройте терминал. Команда `ping 172.16.1.1` должна завершиться успешно, и мы можем попробовать подключиться к нашему облаку по SSH.

```
ssh cirros@172.16.1.1
```

Пароль для образа Cirros по умолчанию — `subswin`. Поздравляю! Теперь у вас есть личное локальное облако. Отсюда мы можем заглянуть в тома [Volumes], вернувшись в веб-интерфейс OpenStack и выбрав эту опцию в меню (предполагая, что мы еще находимся в системе как пользователь demo). Нажмите Create Volume [Создать том] и укажите его имя (без пробелов). Мы можем выбрать размер и не выбирать источник, просто используя пустой том. Нажмите Create [Создать], чтобы OpenStack выделил для вас местечко на диске с помощью Cinder. Теперь мы можем подключить его к своему экземпляру lxf: просто выберите Edit Attachments [Изменить привязки] в меню справа, выберите наш экземпляр из выпадающего меню и нажмите Attach Volume [Подключить том] после выделения имени устройства. Кстати, во время тестирования мы обнаружили, что наша система использовала очередное ожидаемое имя устройства (`/dev/vdb`) вместо предложенного через веб-интерфейс `/dev/vdc`. Видимо, мелкая ошибка. Потом это устройство можно отформатировать и смонтировать как обычный физический диск.

На этом наша краткая экскурсия по ряду базовых функций OpenStack при создании по методу DevStack окончена; но функций, которых мы не коснулись, великое множество. OpenStack — быстро развивающийся проект с массой опций. На следующем уроке мы рассмотрим некоторые продвинутые возможности в более реальном окружении. **LXF**

Как внести свой вклад в OpenStack?

Хороший вопрос! Сообщество OpenStack очень гостеприимно — его участники приложили массу усилий, чтобы дать потенциальным новичкам все необходимое для старта (разумеется, исходный код проекта открыт). В официальной вики (<https://wiki.openstack.org/wiki/HowToContribute>) легко узнать, каким частям проекта требуется помощь и какими методами ее можно оказать. Чтобы ваш код был принят, нужно подписать пользовательское соглашение, но этот процесс обычно не затягивается.

Стоит отметить, что раньше вкладчиком в крупные релизы предоставлялись бесплатные билеты на саммит OpenStack (но увы, не на авиаперелеты). OpenStack несколько необычен не только тем, что код хранится в обычных открытых/публично доступных репозиториях контроля версий, но и тем, что документация проекта и даже инфраструктура для тестирования используют те же методы приема изменений и обновлений. К тому же весь процесс проектирования происходит публично — каждые

полгода сообщество проводит саммит по проектированию для сбора требований и написания спецификации к следующему релизу. Эти саммиты полностью открыты: посетить их может любой желающий. Итогом саммита становится создание плана работ на следующие шесть месяцев. Выборы лидеров каждого проекта тоже проводятся открыто. В OpenStack Foundation любят говорить, что техническое руководство проекта осуществляется по принципу меритократии в обществе, и здесь с ними не поспоришь.

Заметки. Храним и сотрудничаем

Марко Фиоретти учит использовать два инструмента с открытым кодом, чтобы создать личную Evernote-подобную систему для совместной работы онлайн.



Наш эксперт

Марко Фиоретти любит свободное ПО. Он изучает воздействие открытых цифровых технологий на общество и обучает ему.

Интернет способен творить чудеса, позволяя нам контролировать наши задачи и заметки и содействовать сотрудничеству в командах, если, конечно, мы используем правильные инструменты с открытым кодом для этой работы. Два весьма популярных сервиса в этой области — Google Docs и Evernote. Первый (помимо всего прочего) обеспечивает онлайн-хранение, составление календаря и простой сетевой офисный пакет для совместной работы. Evernote утверждает, что хочет помочь вам «организовать вашу жизнь»; это очень мило с их стороны, и в реальной жизни означает использование web-платформы, оптимизированной для быстрого составления заметок, их каталогизирования и присвоения ярлычков всех видов. Вы можете поделиться этими заметками с другими пользователями из основного на браузере сервиса и соответствующего клиента рабочего стола (хотя это придется делать через Wine, поскольку «родная» поддержка Linux отсутствует).

Ранее мы уже говорили, как можно найти более приватные, независимые альтернативы сервисам Google с помощью OwnCloud (Учебники, стр. 72 LXF180 и стр. 78 LXF169). На сей раз мы покажем вам две вещи: как использовать вашу учетную запись Evernote из Linux без браузера, а также когда, почему и как вы можете заменить Evernote (и Google Docs тоже, поскольку все, что вам нужно редактировать онлайн совместно — это простые текстовые документы) свободным ПО, работающим на вашем сервере.

NixNote

Evernote может использоваться любым современным браузером, но многие предпочитают родные клиенты, которые в большей степени интегрированы с остальными программами на их компьютерах, планшетах и смартфонах.

Увы, официального клиента Evernote для Linux нет. С практической точки зрения, на данный момент есть только одна альтернатива с открытым кодом, и это — стороннее приложение

под названием NixNote (бывшее Nevernote, <http://nevernote.sourceforge.net>). На момент написания данной статьи NixNote имеется в двух версиях: 1.6 и 2.0 alpha. Обе версии можно установить и использовать на одном компьютере (не одновременно, конечно), потому что они помещают свои бинарники, базы данных и настройки конфигурации в разные папки. Однако по причинам, которые мы объясним через минуту, у большинства пользователей Linux может быть только одна опция.

NixNote 1.6 имеется в форматах RPM и Deb, а также в общих tar-архивах для 32- или 64-битных дистрибутивов. В последнем случае, распаковав архив, вы должны запустить от имени root включенный в него скрипт под названием `install.sh`. Он поместит надстройку оболочки NixNote в `usr/share/nixnote/nixnote.sh`.

Далее... вы застряли. Теоретически NixNote 1.6 должен быть проще в настройке и работе, и уж конечно, менее рискован в использовании, чем его новая альфа. Но если только вы не работаете на относительно старом дистрибутиве Linux (а это плохо в весьма многих аспектах), вы не сможете синхронизировать свои заметки, поскольку из этой версии NixNote нельзя войти на www.evernote.com. Программа пожалуется, что “SSL support not found [Не найдена поддержка SSH]”, и выключится.

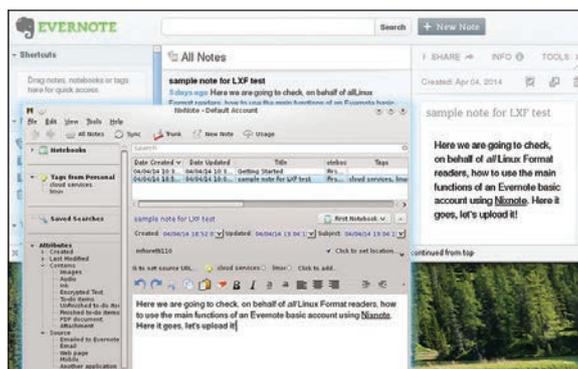
Причина в том, что современные дистрибутивы используют по крайней мере версию 1.0.0 распределенных библиотек `libssl.so` и `libcrypto.so`. A NixNote 1.6 работает только с версией 0.9.8 тех же библиотек, и поскольку программисты NixNote перешли на версию 2, все так и останется.

Решение, предложенное на пользовательском форуме, заключается в установке более старых версий библиотек вместе с новыми, которые вам, естественно, придется сохранить для всех своих других приложений. С технической точки зрения это возможно, но на практике сложно, и если вы не знаете точно, что делаете, пробовать мы бы не советовали.

Если вам не обойтись без клиента рабочего стола Evernote для Linux, лучше попробуйте NixNote 2 alpha (рисунку внизу слева). Насколько мы можем судить, эта альфа уже достаточно хороша для обычных операций и для много другого, если только вы будете постоянно делать резервные копии своих данных (которые будут находиться в папке `$HOME/.nixnote`) и последуете несколькими советам.

Во-первых, поскольку это альфа-версия, NixNote 2 идет с отключенной по умолчанию загрузкой. Чтобы включить ее, снимите галочку с окошка под названием `Disable Uploads to Server` на панели `Edit/Preferences Debugging`. Чтобы заставить Nixnote 2 запускаться, потребуется поработать вручную в командной строке, но это намного менее сложно, чем с NixNote 1.6.

Проблема здесь в том, что NixNote 2 alpha распространяется как один исполняемый файл, с динамическими ссылками



» NixNote 2 хоть и на стадии альфа, но уже работает с Evernote и имеет все основные функции.

А что же такое Node и npm?

Node.js (<http://nodejs.org>) — это платформа JavaScript для быстрых, масштабируемых приложений, которым надо на распределенных устройствах (например, на серверах и ПК в Интернете) обслуживать параллельно множество пользователей.

При подобном сценарии чистая вычислительная мощность привлеченных CPU значит меньше, чем способность программы обрабатывать в двух

направлениях множество одновременных, но полностью независимых асинхронных потоков данных. Node.js в состоянии весьма достойно с этим справиться, обрабатывая все входящие и исходящие данные с помощью неблокирующей системы, управляемой событиями.

Многие приложения Node.js не созданы с нуля, а собраны из повторно используемых блоков кода,

которые именуются Node Packaged Modules. Основной, официальный репозиторий для этих модулей — <http://npmjs.org>. Node Package Module, чей исполняемый файл именуется *npm*, ищет пакеты в онлайн-репозиториях и устанавливает модули и их зависимости примерно так же, как *yum*, *apt* и их графические интерфейсы для обычных пакетов Linux.

на определенные библиотеки. Не пугайтесь: «динамические ссылки» просто означают, что вместо включения целиком необходимых ему библиотек файл содержит только названия мест, где он может их найти.

И если ваш дистрибутив Linux содержит эти файлы, но под немного другими именами или/и в других папках, то исполняемый *NixNote 2* начнет жаловаться — да и выключится. Решение заключается в том, чтобы найти эти библиотеки и создать в папках символические ссылки на них, с правильными именами. Например, в Fedora следует создать три ссылки, от имени *root*:

```
#> ln -sf /usr/lib64/libssl.so.1.0.0k /usr/lib64/libssl.so.1.0.0
#> ln -sf /usr/lib64/libcrypto.so.1.0.0k /usr/lib64/libcrypto.so.1.0.0
#> ln -sf /usr/lib64/libpoppler-qt4.so.3.7.0 /usr/lib64/libpoppler-qt4.so.4
```

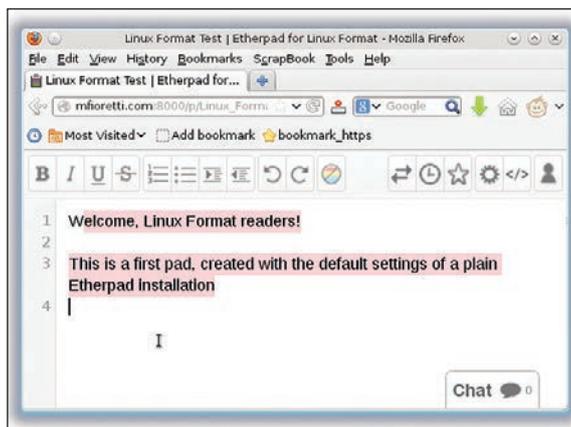
Найдя все библиотеки на положенных местах, *NixNote* откроет окно, где вы должны будете войти на www.evernote.com и авторизовать приложение (по умолчанию, сроком на год), чтобы удалять или редактировать существующие заметки, создавать новые и присваивать им ярлыки или упорядочивать их в блокноте.

Интерфейс очень простой и симпатичный, и очень многообещающий, несмотря даже на то, что пока работают не все функции. Например, шифрование заметок, так же, как (опционально) автоматическая синхронизация с www.evernote.com на запуске и выключении, пока не работают. Помимо основных инструментов форматирования, редактор *NixNote 2* также имеет поддержку контрольных списков и таблиц. Есть возможность отобразить подключенные файлы PDF и настроить форматы даты и времени. Вы можете присвоить своим заметкам как URL, которые легко запомнить, так и расположение: просто щелкните дважды по стрелке вправо от меню Notebook и затем по Click to Set Location, чтобы ввести любые координаты, какие захотите.

Способ *NixNote* работать с заметками, ярлыками, а самое главное, блокнотами с распределенным доступом, несколько отличается от образа действий других клиентов Evernote. Частично причина кроется в том, что это — результат неофициальной, добровольной разработки с меньшим количеством ресурсов, а частично в том, что это осмысленный выбор иного подхода.

Например, в *NixNote* личные блокноты и блокноты с распределенным доступом появляются в одном списке. Ярлыки, ссылающиеся на блокнот с распределенным доступом, выглядят так же, как и любой другой ярлык, но использоваться могут только в этом блокноте. Нельзя присвоить любой из своих ярлыков заметке в блокноте с распределенным доступом. А перемещать заметки из блокнота с распределенным доступом в блокнот с нераспределенным доступом можно, только если они новые — то есть в блокноты, которые еще не были синхронизированы с www.evernote.com.

И последнее, что вам нужно знать о *NixNote* — программа умеет работать с разными, независимыми учетными записями,



но не может делать этого одновременно. Вам придется работать над ними параллельно, запуская программу в командной строке дважды, каждый раз передавая имя используемой учетной записи:

```
nixnote --accountId=<id1> &
nixnote --accountId=<id2> &
```

Etherpad

NixNote, конечно, мил, но это все же клиент Evernote. И как таковой, он не может решить две больших проблемы, крайне важные для многих пользователей. Самая очевидная — это конфиденциальность и собственность на данные: он не задумается разместить ваши личные заметки на чьем-то чужом сервере, если вы разместили их на своем. Еще одно очевидное ограничение как Evernote, так и Google Docs в том, что они работают только там, где есть доступ к широкополосному интернету. А как же тогда быть сельским школам, НГО и подобным организациям, чьи пользователям нужно делиться заметками в локальной сети, поскольку это единственный вид связи, который им нужен или вообще доступен?

Etherpad (который иногда именуется *Etherpad-lite* — <http://etherpad.org>) является хорошим решением с открытым кодом для всех подобных сценариев. Скорейший, хоть и немного несправедливый способ определить эту программу — сказать, что это урезанная смесь Evernote и Google Docs. Однако более точным определением будет «сетевой редактор для совместной работы в реальном времени». Это означает, что вы и ваши сотрудники можете писать разные абзацы одного документа одновременно, и при этом в браузере разными цветами будет отображаться то, что делают другие. Встроенный чат-клиент позволяет сотрудникам обсуждать свои действия. При необходимости щелчок по кнопке Playback покажет всю последовательность редактирования до текущей заметки. Интерфейс для каждого пользователя

» Установка сложная (относительно), однако после ее завершения совместное редактирование онлайн будет быстрым и легким.

Скорая помощь

NixNote работает с блокнотами с распределенным доступом и присваивает ярлыки не так, как другие клиенты Evernote. Загляните в онлайн-документацию, чтобы не было сюрпризов.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

автоматически настраивается на язык по умолчанию браузера этого пользователя.

Простой способ обзавестись собственным *Etherpad* — установить комплекс *Etherpad* (www.turnkeylinux.org/etherpad) на виртуальный сервер. Это готовый к использованию образ основного дистрибутива Linux, в котором уже загружена и настроена последняя версия *Etherpad* и все необходимое для ее работы с самой первой минуты, включая сервер *MySQL* с правильным пользователем и базой данных и почтовый сервер *Postfix* для отправки уведомлений по электронной почте. Единственный недостаток этого комплекса в том, что он требует размещения на собственном, отдельном (виртуальном или физическом) оборудовании, на котором больше не должно быть ничего.

Установка *Etherpad* на ваш собственный сервер хотя и возможна даже в том случае, когда вы не являетесь профессиональным web-мастером или системным администратором, но имеет более строгие требования и более сложна, чем настройка *WordPress*, *Digital* или даже *Owncloud*. Первая причина — память (см. врезку «Проблемы с памятью» внизу) и архитектура других программ.

Все эти другие программы — приложения LAMP. Это наборы динамических web-страниц, написанных на языке PHP, которые хранят свои данные в базе данных *MySQL* и исполняются web-сервером общего назначения (в Linux — обычно *Apache*, которому и обязана буква А в LAMP). Чтобы установить и запустить его, нужен только доступ к сетевому пространству и база данных *MySQL*. *Etherpad* же является независимым web-сервером, написанным почти полностью на JavaScript с помощью платформы *Node.js/npm* (см. врезку «А что же такое Node.js и npm?», стр. 79). По этой причине для большинства дистрибутивов Linux официальные двоичные пакеты или пакеты отсутствуют. Во всех случаях для установки *Etherpad* на сервер Linux нужен доступ к терминалу от имени *root*. По всем этим причинам точные под-

робности и последовательность команд различаются в разных дистрибутивах. Есть специальные руководства для некоторых дистрибутивов (см. «Документация», стр. 81), но они не всегда свежие.

Поэтому мы предлагаем вам обобщенное описание основных шагов и наиболее вероятных проблем, с которыми вы можете столкнуться. Это должно помочь вам быстрее разобраться с официальной и самой базовой документацией, подсказывая, куда обратиться, если что-то не заладится.

Подготовка среды

Если на вашем сервере еще нет, как минимум, пакетов *git*, *curl*, библиотек разработки *OpenSSL*, компилятора *GCC*, утилиты *Make*, *Node.js* и *npm*, установите их (используя свой менеджер пакетов и репозитории своего дистрибутива; возможно, для двух последних придется авторизовать дополнительный репозиторий — например, добавить *ppa:chris-lea/node.js*, чтобы получить самую последнюю версию *Node.js* для Ubuntu). Это необходимо, потому что получить текущую версию *Etherpad* означает скачать ее исходники с помощью системы контроля версий *Git*, а затем скомпилировать и/или настроить их части с помощью *GCC* и *Make*.

Вы также должны создать базу данных *MySQL* и пользователя, зарезервированного для *Etherpad*, а также пользователя Linux, единственной целью которого будет установить и запустить программы с этим или эквивалентным ему комментариями. В приведенных ниже примерах команд имя этого пользователя — *etherpad*, но оно может быть любым.

Еще одно обязательное действие (поскольку *Etherpad* не использует стандартный web-сервер, но должен иметь свой отдельный порт TCP) — открытие порта в брандмауэре. В системах CentOS/Red Hat, используемых на нашем уроке, это делается посредством добавления к файлу */etc/sysconfig/iptables* такого правила:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8000 -j ACCEPT
```

Вся подготовительная работа, которую мы здесь описали, должна была выполняться от имени *root*. Чтобы осуществить саму установку, надо переключиться на нового пользователя и скачать *Etherpad* с помощью *Git*:

```
#> su - etherpad
#> cd /home/etherpad
#> git clone git://github.com/ether/etherpad-lite.git
```

В этот момент наконец становится возможно подготовить файл настройки и запустить скрипт исходной установки. Чтобы выполнить первый шаг, перейдите в директорию *etherpad-lite*, созданную *Git*, скопируйте файл *settings.json.template* в *settings.json* и откройте этот последний в текстовом редакторе. Как минимум, вам придется настроить параметры *MySQL* и изменить заглавие, TCP-порт и переменную *sessionKey* на соответствующие показатели:

```
"title": "Etherpad for Linux Format",
...
"port": 8000,
...
"sessionKey": "LXF - отличный журнал"
```

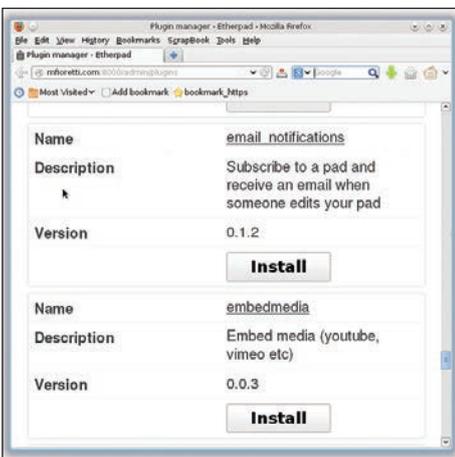
Если вы собираетесь пользоваться web-интерфейсом администрирования (описанным далее), вам также следует настроить в начале раздела *users* пользователя *admin*:

```
"users": {
```

Скорая помощь

Как обеспечить возможность совместного редактирования, если вам требуется конфиденциальность ваших заметок или у вас нет широкополосного соединения? Конечно, с помощью *Etherpad*!

➤ Разделы *Plugins* web-интерфейса администрирования *Etherpad*: довольно скромные, но функциональные.



Проблемы с памятью

Etherpad сильнее нагружает память по сравнению с другими системами управления контентом. Ставя его на виртуальном частном сервере (VPS), попробуйте, чтобы на том было не менее 256 МБ. Это очень важно, иначе не исключено, что вам не удастся даже запустить *Etherpad* сразу после установки.

Основная причина в том, что *Etherpad* использует *npm* для нахождения и установки обновлений

и опциональных модулей. Но для этого копия *npm*, упакованная в *Etherpad*, должна создать локальный индекс.

Создание этого индекса с нуля, вероятно, будет самой ресурсоемкой задачей, с которой придется иметь дело установке *Etherpad* для малых групп, использующих эту систему. Вы увидите, что страница администрирования *Etherpad* может зависнуть,

а в командной строке или журнале у вас будут ошибки такого рода:

```
./bin/run.sh: line 38: 16471 Killed node $SCRIPTPATH/node_modules/ep_etherpad-lite/node/server.js $*
```

Это почти наверняка означает отказ *npm* или прекращение его работы из-за недостатка памяти. Единственным решением будет увеличить память на вашем сервере.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

```

"marco": {
"password": "lxf",
"is_admin": true
},

```

Из директории **etherpad-lite**, и по-прежнему от имени пользователя **etherpad**, запустите следующие команды:

```

#> ./bin/installDeps.sh
#> ./bin/run.sh

```

Первый скрипт проверяет, все ли зависимости на месте, а второй запускает сервер. Если все в порядке, то когда вы запустите его, ваш собственный сервер *Etherpad* будет доступен через ваш браузер на сайте <http://yourwebsite.com:8000>.

В отличие от Evernote, *Etherpad* не имеет готового к использованию обычным пользователем графического клиента рабочего стола. Здесь есть приложения командной строки вроде *Ethersync* (<https://github.com/payload/ethersync>) и *Etherpad streams* (<https://www.npmjs.org/package/etherpadstream>), которые позволяют редактировать заметки из командной строки или загружать в *Etherpad* выходные данные какого-то скрипта; но они вряд ли подойдут всем. Однако базовый интерфейс *Etherpad* достаточно прост, чтобы любой пользователь смог освоиться в нем после установки и начала работы. Самая важная функция, о которой стоит упомянуть, это кнопка с угловыми скобками: щелкните по ней, и получите код HTML, который вы сможете скопировать и вставить на любую другую веб-страницу, чтобы все могли видеть и редактировать ваши записи.

Запуск Etherpad и работа с ним

Как и его установка, обновления *Etherpad* работают иначе, чем те, к которым вы, вероятно, привыкли с другими системами управления контентом [Content Management Systems, CMS]. Чтобы проверить наличие новых версий и скачать их, вы должны запустить такую команду **git**:

```

#> git pull origin

```

Когда вы это сделаете, следующее исполнение **run.sh** действительно произведет обновление. Что касается автоматического (пере)запуска при каждой загрузке, *Etherpad* работает более или менее так же, как другие сервисы Linux/Unix: вы должны скачать, настроить при необходимости и включить скрипт оболочки для вашего дистрибутива, следуя инструкциям на wiki Github <http://bit.ly/1sZpBYo>.

Запустить несколько независимых *Etherpad* на одном сервере с той же установкой тоже можно, и вот как это делается.

» Назначьте отдельный порт TCP для каждого экземпляра и настройте брандмауэр на вашем веб-сервере, чтобы принимать соответствующие соединения.

» Создайте каждому экземпляру по файлу настройки **settings.json**.

» В скриптах **init** и во всех других местах, где они могут вам понадобиться, замените вызовы **run.sh** одним вызовом *Etherpad*, каждый с собственным файлом настроек:

```

run.sh -s work_etherpad_settings.json

```

```

run.sh -s family_etherpad_settings.json

```

Из-за своей архитектуры *Etherpad* имеет два свойства, которые иногда способны создать проблемы. Одна — его URL'ы по умолчанию, чьи номера портов могут запутать неопытного пользователя, особенно если на одном и том же сервере имеется более одного *Etherpad*. Другая — контроль доступа на уровне сайта: используя нужные плагины, вы можете заставить своего пользователя вводить пароль для чтения и/или редактирования некоторых заметок, но по умолчанию любой сайт *Etherpad* является публичным.

Обе проблемы можно решить, «скрыв» *Etherpad* за сервером *Apache*, который будет обрабатывать аутентификацию и автоматически переформатировать URL для пользователей. Этот трюк,

Документация

Немалый объем документации *NixNote* доступен в виде документов Evernote на www.evernote.com/pub/baumgarr/evernote. Здесь есть FAQ, инструкции по установке, подробное руководство пользователя и сведения для разработчиков. Для *Etherpad*, помимо документации на сайте, мы бы советовали

прочитать руководство на Digital Ocean (<http://bit.ly/1p2yO2u>) для CentOS, и в wiki *Etherpad Lite* (<http://bit.ly/1m0M8jt>) для Ubuntu и вообще для производных Debian. Оба предоставляют все необходимые подробности для каждого дистрибутива, которые не вместились в этот материал.

называемый «обратный прокси [reverse proxy]», объясняется в Reverse Proxy Howto на wiki *Etherpad* (<http://bit.ly/1guEg59>).

Некоторые задачи по администрированию *Etherpad* можно выполнять и без командной строки, если вы включили в файл настройки пользователя-администратора. Направьте свой браузер в подпапку **/admin** своего *Etherpad*, введите пароль, и перед вами появится весьма спартанского вида страница с тремя разделами: Settings [Настройки], Troubleshooting [Поиск неисправностей] и Plugin manager [Менеджер расширений]. Первый — это просто текстовая область, где можно просматривать и редактировать файл **settings.json**. Второй описывает полную конфигурацию вашей установки.

Официальная страница (<http://bit.ly/1iMF8np>) перечисляет плагины почти для всех задач и на все вкусы, от дополнительных функций форматирования до списков задач и графических тем. По умолчанию *Etherpad* может только импортировать или экспортировать заметки в два формата — текстовый и HTML, но есть плагины для конвертирования во множество других форматов через *AbiWord*. Другие расширения позволяют администратору управлять группами пользователей или (как вы можете видеть на рисунке ниже) встраивать мультимедиа-контент в ваши заметки.

Plugin manager позволяет добавлять все эти функции в *Etherpad* несколькими щелчками, однако это не всегда наилучший способ справиться с этой задачей. Установка командной строки через *npm* использует следующий простой синтаксис:

```

npm install ep_<plugin-name>

```

и дает тот же результат, и будет намного удобнее, если вам нужно настроить более одного *Etherpad*. Обновление и удаление плагинов работает так же:

```

npm install --upgrade ep_<plugin-name>

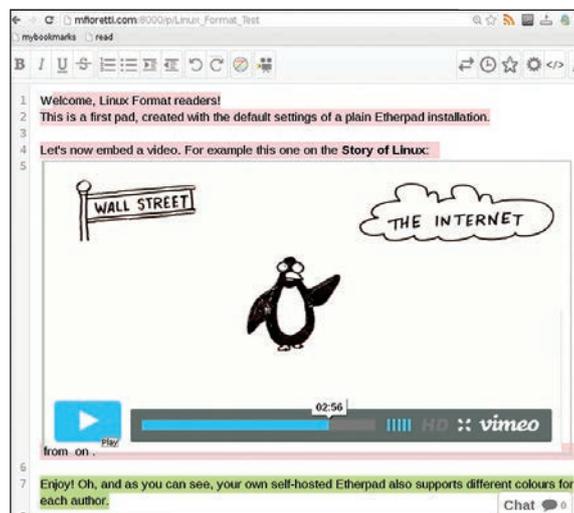
```

```

npm uninstall ep_<plugin-name>

```

Во всех случаях не забывайте, что по окончании надо перезапустить *Etherpad*. **LXF**



Скорая помощь

По сравнению с другими CMS *Etherpad* очень прожорлив в плане памяти! Не забудьте проверить, сколько ОЗУ вы можете предоставить!

» Правильные плагины могут сотворить с *Etherpad* настоящие чудеса, от конвертирования ваших заметок во множество форматов до встраивания мультимедиа.

Введение в Python

Изучите основы языка программирования Python: **Даниэль Сэмюэлс** покажет вам, за какие веревки дергать.



Наш эксперт

Даниэль Сэмюэлс начал учиться программировать в 14 лет и работал платным программистом еще в школе. Ныне он ведущий разработчик в Onepascalmedia, творческом цифровом агентстве, которое находится в Кембридже.

➤ Вот так ваша оболочка Python должна выглядеть при запуске примеров про строки.

На этом уроке мы рассмотрим некоторые из основных понятий программирования и как они используются в приложениях Python. Вы, наверное, слышали о Python раньше: это чрезвычайно популярный в технологическом сообществе язык программирования. И частные лица, и организации используют его для больших и малых проектов — и на настольных ПК, и в Интернете.

Популярностью он обязан своему простому и легко изучаемому синтаксису, стандартной библиотеке, содержащей почти все утилиты, которые вам в жизни понадобятся, и вспомогательному репозиторию библиотек для всего остального. Python способен работать на большинстве операционных систем, и дистрибутивы Linux обычно поставляются с предустановленным Python или делают его доступным через свои менеджеры пакетов.

Чтобы начать писать на Python, откройте терминал и введите **python**. Если Python установлен правильно, запустится интерактивная оболочка; это полное окружение Python, которое позволяет писать и запускать любой код Python таким же образом, как с файлом .py. Если вы не получите оболочку Python, может потребоваться загрузить и установить ее с сайта Python — www.python.org — или через менеджер пакетов вашего дистрибутива.

Пишем свои первые строки кода

Итак, Python у нас заработал, можно начать писать код! Пойдем обычным путем и попробуем напечатать некоторый текст — введите следующую команду: **print "Hello World!"** и нажмите Enter, чтобы она сработала. Вы должны увидеть строку текста (известную как 'string [строка]'), говорящую Hello World! Этот текст

можно заменить почти любым другим, и Python будет печатать его на экране; что и является основой большинства приложений Python. Далее мы попробуем начать сохранять какие-нибудь данные, для чего воспользуемся штукой под названием «переменная»: это просто способ именования того, что вы можете снова получить позже. Чтобы определить переменную, нужно просто задать ей имя и какое-нибудь значение. Вот пример переменной `fruit` и ее использования:

```
fruit = 'Банан'
```

При задании переменной никакого вывода на экран вы не увидите, но можно послать его на печать и отобразить название вашего фрукта (примечание: строки с префиксом '>>' показывают, какой ожидается вывод):

```
print fruit
```

```
>> Банан
```

Python позволяет выполнять множество действий со строками: например, найти длину строки, сосчитать количество вхождений какой-либо буквы, изменить регистр или просто взять фрагмент строки. Для иллюстрации, посмотрите на следующие примеры:

```
len(fruit)
```

```
>> 5
```

```
fruit.count('a')
```

```
>> 2
```

```
fruit.upper()
```

```
>> БАНАН
```

```
fruit[0:3]
```

```
>> Бан
```

```
danielsamuels@ubuntu: ~
danielsamuels@ubuntu:~$ python
Python 2.7.3 (default, Apr 10 2013, 05:46:21)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> age = 18
>>>
>>> if age >= 18:
...     print "You are allowed to enter."
... else:
...     print "You are not allowed to enter."
...
You are allowed to enter.
>>> |
```

Мы также можем совместить печать строк и переменных для получения индивидуального вывода — скажем, объявив, какой наш любимый фрукт:

```
print "Мой любимый фрукт - " + fruit
>> Мой любимый фрукт - Банан
```

С помощью специальной функции, называемой `format`, можно также вставлять переменные внутри наших строк для обеспечения более динамичного вывода текста. Этим можно воспользоваться, чтобы поприветствовать пользователя вашего приложения, сказав 'Здравствуй, <имя>.', или показать цену товара в интернет-магазине:

```
print "Я купил {} в магазине.".format(fruit)
>> Я купил Банан в магазине.
```

Переменные пригодятся не только для создания и форматирования строк текста: их также можно применить в математических вычислениях, для сохранения данных с целью последующего использования, для принятия решений и целого ряда других вещей. Вот еще несколько примеров использования переменных:

```
number1 = 10
number2 = 5
number1 + number2 # Сложить 2 числа.
>> 15
number1 * number2 # Перемножить 2 числа.
>> 50
number1 ** number2 # Возвести number1 в степень number2
>> 100000
fruit = "Банан"
fruit * 3 # Повторить строку 'Банан' 3 раза.
>> БананБананБанан
```

Последний пример подчеркивает, что Python не делает умножение, как если бы переменная `fruit` была числом, а повторяет 'Банан' три раза, зная, что это строка.

Логика в приложении

Логика — главная часть программы; важно бывает знать, когда показывать имя пользователя на web-странице или кнопку в настольном приложении. Простейший способ ввести логику в Python, да и в большинство языков программирования — использовать утверждение `if` [если]. Определив, что мы хотим проверить, мы можем запускать тот или иной код в зависимости от результата проверки, что позволяет нам контролировать ход своего приложения. Например, вы можете отображать некоторое содержание только для пользователей старше определенного возраста, и несложное утверждение `if` добьется этого с легкостью.

```
age = 18
```

Ваши дальнейшие шаги

Всякие штуки, с которыми мы сегодня познакомились, не являются специфическими для Python: почти в каждом языке программирования есть переменные, строки, функции и методы, хотя они не обязательно записываются таким же образом. Желая узнать о Python побольше мы бы советовали посетить официальный сайт <https://wiki.python.org/moin/BeginnersGuide>. Там имеются

страницы со ссылками на другие полезные ресурсы, способные поднять ваши навыки в программировании на новый уровень: мы здесь коснулись его лишь поверхностно. Любителям учиться программировать в интерактивном режиме советуем также Codecademy (www.codecademy.com/tracks/python), их урокам очень легко следовать.

```
if age >= 18:
    print "Вы можете войти."
else:
    print "Вам войти нельзя."
>> Вы можете войти.
```

Возможно, вам придется нажимать `Enter` дважды, чтобы утверждение сработало. Как видите, мы проверяем, что наша переменная `age` [возраст] больше или равна 18; если да, то заходите, а в противном случае войти нельзя.

Использование файлов

Одна из проблем оболочки Python — то, что не очень легко сохранить сделанную работу для дальнейшего употребления, так что воспользуемся PY-файлом (файлом с расширением `.py`), чтобы сберечь свои труды. Выйдите из оболочки Python, введя `exit()` и нажав `Enter`, затем откройте свой любимый текстовый редактор и сохраните где-нибудь пустой файл с именем `tutorial.py`. В терминале перейдите в папку, где помещен ваш новый PY-файл. Теперь можно вносить изменения в PY-файл в текстовом редакторе, а затем просто запускать `python tutorial.py` в терминале, чтобы код работал. Попробуем: наберем в файле `tutorial.py` текст

```
print "Мы теперь используем файл .py!"
```

а затем в терминале —

```
python tutorial.py
>> Мы теперь используем файл .py!
```

Пойдем дальше: запишем некоторые из своих переменных, строк и утверждений `if` в вашем новом PY-файле и посмотрим, что произойдет, когда файл запустится в терминале; можете взять тот код, что мы уже написали. Вы заметите, что при запуске PY-файла заодно генерируется файл `.pyc`, но его опасаться нечего: Python просто заносит туда ваш код. **LXF**

```
danielsamuels@ubuntu: ~
danielsamuels@ubuntu:~$ python
Python 2.7.3 (default, Apr 10 2013, 05:46:21)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> fruit = 'Banana'
>>> print fruit
Banana
>>> len(fruit)
6
>>> fruit.count('a')
3
>>> fruit.upper()
'BANANA'
>>> fruit[0:3]
'Ban'
>>> print "My favourite fruit is a " + fruit
My favourite fruit is a Banana
```

➤ Запуск в оболочке кода по-прежнему тоже работает, но управлять им становится трудно.

Назад к основам: Пути Python

Нейл Ботвик идет по единственно верному пути обработки кросс-платформенных файлов в Python.



Наш эксперт

У Неила Ботвика по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не расскажет, где находится центральный сервер.

➤ **Функция walk()** позволяет нам делать все, что за благо рассудится, со всей иерархией директории.

Создание скриптов Python для работы на вашем собственном компьютере — одно дело: вы знаете, что где находится. Однако если вы хотите создать нечто более мобильное, придется задуматься о расположении файлов и о путях к ним. А вот когда вы начнете рассматривать кросс-платформенную совместимость, особенно совместимость с Windows, все станет действительно интересным. По какой-то причине Microsoft решил перевернуть — в буквальном смысле — освященную веками (почти) привычку разделять директории слэшем. Употребив в MS-DOS обратный слэш, они не только навлекли на дикторов телевидения или радио обязанность применять нелепый термин «прямой слэш [forward slash]» при озвучивании URL, но и успешно разрушили множество попыток создания кросс-платформенных программ. Между прочим, предполагается, что Windows принимает любой символ в качестве разделителя директорий, но использование нужного слэша в лучшем случае может оказаться непредсказуемым.

Единственно верный путь?

Вряд ли кого-то удивит информация, что в Python имеется модуль для борьбы с этим, под названием `os.path`. Это часть модуля `os`, и если этот последний у вас уже есть, вам незачем загружать `os.path` отдельно. Если же вам нужны только функции работы с путями, можно импортировать только эту часть:

```
import os.path
```

Прежде чем перейти к обратным и прямым разделителям и прочим Windows'измам, давайте рассмотрим более общие и полезные функции. Чтобы улучшить читаемость, мы в большинстве примеров опустили префикс `os.path`, поэтому, например, там, где стояло `isfile()`, надо использовать в скриптах `os.path.isfile()`.

Когда вы принимаете имя пути от пользователя, первым делом надо проверить, действует ли оно; это делается с помощью функций `exists()`, `isdir()` и `isfile()`, которые проверяют именно то, что и предполагают их имена. Эти функции следуют символическим ссылкам, и если вам важно, что указанный вам путь — на самом деле файл, а не symlink, следует использовать нечто вроде

```
if isfile(path) and not islink(path)
```

Поскольку `exists()`, `isfile()` и `isdir()` следуют символическим ссылкам, они вернут False, если ссылка будет битой. Если вы используете `lexists()`, она вернет True для любого файла, директории или symlink, даже для битых ссылок.

Работа с путями

Итак, мы убедились, что путь существует, и у нас есть несколько функций, чтобы с ним работать. Обычно эти функции — строковые процессоры, которые используют обычные правила пути. Они не обращаются к самим файлам, поэтому вам надо сначала проверить, существуют ли они. Начнем с `basename()`, которая возвращает последнюю часть пути, и `dirname()`, которая возвращает

остальное.

```
>>> basename('aa/bb/cc')
```

```
'cc'
```

```
>>> dirname('aa/bb/cc')
```

```
'aa/bb'
```

При перемещении по дереву файловой системы бывает удобнее работать с абсолютными путями — `isabs()` сообщит нам, является ли данный путь абсолютным, то есть таким, который начинается со слэша в Linux и обратного слэша или буквы диска в Windows. Или можно использовать `abspath()`, чтобы конвертировать путь в абсолютный:

```
>>> isabs('aa/bb')
```

```
False
```

```
>>> abspath('aa/bb')
```

```
'/home/nelz/aa/bb'
```

```
>>> abspath('/home/nelz/aa/bb')
```

```
'/home/nelz/aa/bb'
```

Вы здесь видите, что использование `abspath()` на абсолютном пути абсолютно ничего не меняет (уж извините). Противоположностью `abspath` является `relpath()`, которая конвертирует абсолютный путь

```
File Edit View Bookmarks Settings Help
File Edit Options Buffers Tools Complete In/Out Signals Help
#!/bin/env python
import os.path, sys
def countfiles(_, dirname, filelist):
    print "%s contains %d files" % (dirname, len(filelist))
os.path.walk(sys.argv[1], countfiles, None)

UU:~:~:~F1 walk.py All L1 (Python)
~/PreviousIssues/24 contains 17 files
~/PreviousIssues/53 contains 3 files
~/PreviousIssues/53/DVD contains 7 files
~/PreviousIssues/53/Covers contains 6 files
~/PreviousIssues/53/CD contains 7 files
~/PreviousIssues/59 contains 3 files
~/PreviousIssues/59/Covers contains 9 files
~/PreviousIssues/59/DVD contains 11 files
~/PreviousIssues/59/CD contains 7 files
~/PreviousIssues/131 contains 2 files
~/PreviousIssues/131/Answers contains 31 files
~/PreviousIssues/131/KDE44 contains 8 files
~/PreviousIssues/146 contains 2 files
~/PreviousIssues/146/Answers contains 32 files
~/PreviousIssues/146/WaysToGetTheMostFromOpenSource contains 6 files
nelz@hactar LinuxFormat/CodingConcepts 0%
UUU:~:~:~F1 *shell* Bot L524 (Shell:run)
```



в путь, относящийся к текущей директории, или к другой директории — при наличии второго аргумента. В предыдущем уроке мы использовали переменную среды `$HOME` для обнаружения домашней директории пользователя. Мы также можем это сделать с помощью `expanduser()`.

```
>>> expanduser('~')
'/home/nelz'
>>> expanduser('~fred')
'/home/fred'
```

При использовании для текущего пользователя, как в первом примере, функция читает `$HOME`, просто избавляя нас от хлопот делать это напрямую.

Создание путей

Создание путей из отдельных компонентов может оказаться несколько запутанным:

```
path = root + '/' + dir + '/' + subdir + '/' + file
```

работает, однако это не отличается элегантностью и предполагает, что ни одно из имен директории не включает в себя завершающего `/`. Функция `join` в `os.path` позаботится обо всем этом.

```
path = os.path.join(root, dir, subdir, file)
```

Здесь пути соединяются вместе, причем обеспечен в точности один разделитель директорий между каждой парой элементов. Вы должны знать, что если какой-то элемент начинается с `/`, путь считается абсолютным, а все предыдущие элементы отбрасываются. Поскольку слэш ставится только между элементами, пример возвращает `root/dir/subdir/file`. Если вам нужно создать переменную пути, заканчивающуюся слэшем (что часто излишне, поскольку при желании добавить больше элементов можно опять использовать `join`), просто поставьте последним элементом пустую переменную:

```
>>> join('root', 'dir', 'subdir', '')
'/root/dir/subdir/'
```

Но как бы вы ни были осторожны при создании и работе с путями, в итоге все равно можно получить всякую гадость типа `../home/fred/movies/./pictures`, особенно принимая пути от пользователя. Мы уже видели, как `abspath()` конвертирует относительные пути в абсолютные; заодно она еще и разбирается с подобной ерундой. Ну, а если вам не нужен абсолютный путь, но и бессмысленный путь тоже не нужен? Используйте `normpath()`, чтобы избавиться от сомнительных слэшей и блуждания по директории. Вы также можете прояснить путь с помощью `realpath()`, который также разрешает символические ссылки, однако у `normpath` и `abspath` припасен еще один трюк — они конвертируют разделители директории в форматы Windows и POSIX. Запуская свои скрипты в Linux, разницы вы не заметите.

Слэш и обратно

Пока что все это относилось в основном к Linux (ну, если быть предельно точными, то к POSIX). Некоторые из этих функций работают в Windows, но не те, которые обращаются к файловой системе, недоступной в Windows, например, к `symlinks`. Функции `normpath()` и `abspath()` также конвертируют форматы пути в форматы их родной операционной системы, поэтому запуск

```
normpath('Documents/Letters')
```

в Windows даст вам `Documents\Letters`, но оставит все без изменений в системах POSIX. Так что всегда используйте пути POSIX в своих скриптах, и пусть `normpath()`, `abspath()` и `join()` заботятся обо всех необходимых переводах, когда ваш скрипт будет запускаться где-то еще.

Еще одно различие между разными операционными системами — заглавные и маленькие буквы; в файловых системах Linux заглавные буквы важны, а в Windows — нет. Решение — работать с путями, где заглавные буквы учитываются, коли уж вы к этому привыкли, и использовать функцию `normcase()` для их перево-

Модули для отдельных ОС

Модули `os.path` — это нечто вроде надстройки: существуют разные модули пути для разных операционных систем, например, `posixpath`, `ntpath`, `macpath` и `os2emxpath`.

Нужный для размещающей операционной системы модуль загружается `os.path`, но вы можете использовать модули непосредственно, если, например, хотите работать с путями Windows в системе Linux.

да. Она оставляет путь нетронутым в системах, где учитываются заглавные буквы, и конвертирует их в маленькие буквы в тех системах, где регистр неважен. Вы должны знать, что это определяется проверкой операционной системы, где запущен скрипт; подобно многим из этих функций, он просто работает со строкой, а файловой системы не затрагивает. Об этом следует помнить, если есть хоть какая-то вероятность того, что ваш скрипт может использоваться в файлах в файловой системе, нечувствительной к регистру, смонтированной на операционной системе POSIX, например, при работе с фотографиями с форматированной в FAT карты SD.

Эта функция также позволяет быстро определить, работает ли ваш скрипт в Windows:

```
if normcase('XyX') == normcase('xyz'):
    print "К сожалению, у вас Windows"
```

Идем своим путем

Хотя модуль именуется `os.path`, он не только работает с путями. Хотя большая часть операций, которые мы рассмотрели, в основном являются строковыми операторами со специальными способностями, необходимыми для работы с путями, есть также функции, которые обращаются к файловой системе, например, `exists()` и ее друзья. Эти функции работают только с метаданными файловой системы — они получают доступ к информации о файлах, но получают ее не из файлов. Сюда входят обычные метаданные, которые могут быть вам интересны: `getatime`, `getmtime` и `getsize` делают именно то, о чем говорят их имена. Если вы хотите сменить владельца или разрешения, вам нужны функции из модуля `os`; такие, как `os.chmod`, `os.getuid` и `os.setuid`.

Модуль `os` также используется для чтения содержимого директории, с помощью `os.listdir`, но `os.path` обладает мощной функцией работы с директориями, по крайней мере, в серии 2.x: `walk()`. Она берет директорию и функцию в качестве аргументов и применяет эту функцию к директории и любым директориям ниже — то есть прочесывает дерево директорий, используя эту функцию. Это объясняется в примере:

```
#!/bin/env python
import os.path, sys
def countfiles(_, dirname, filelist):
    print "%s contains %d files" % (dirname, len(filelist))
    os.path.walk(sys.argv[1], countfiles, None)
```

Для каждой повстречавшейся директории `walk` вызывает функцию с тремя аргументами. Второй — имя директории, третий — список, содержащий все файлы и поддиректории в этой директории (созданный с помощью `os.listdir`). Первый аргумент определяется пользователем; он позволяет передать все, что требуется, из `walk` в функцию. Аргумент должен быть задан, хотя мы его здесь не используем; мы назвали его `underscore` — это условное обозначение в Python имен, которыми вы не будете пользоваться. Вызывая `walk()`, вы также даете ему три аргумента: первый — это путь, по которому вы должны пройти, второй — имя функции, и третий — аргумент, который вы передали этой функции; здесь стоит `None` [ничего], потому что мы его не используем. Обратите внимание, что в Python 3 эта функция перешла в `os.walk()`, и синтаксис другой, без третьего аргумента. **LXF**

Python: Освоим Minecraft Pi

Джонни Бидвелл ставит Python на свой Pi, чтобы порезвиться в *Minecraft*.



Наш эксперт

Джонни Бидвелл желает, чтобы по-больше вещей делалось из блоков.

Пожалуй, щедро предоставляемая *Minecraft: Pi Edition* будет повеселее щедро предоставляемой *Wolfram Mathematica: Pi Edition!* Это урезанная версия популярной карманной редакции *Pocket Edition*, и поэтому в ней нет опасных для жизни перипетий, но зато блоков — палками не забудешь, плюс три типа побегов, с которых можно этих палок наломать.

Другими словами, есть масса инструментов для раскрепощения вашего творчества, но... как же все эти щелчки утомляют! А вот с помощью редакции, включающей эlegantное API Python, вы сможете воплотить в жизнь блочные версии своих самых необузданных мечтаний всего несколькими строками кода.

Если ваш Pi уже настроен и запущен, первым делом загрузите с сайта <http://pi.minecraft.net> свежую версию программы, в свой домашний каталог. Авторы рекомендуют использовать Raspbian, и мы к ним присоединимся — с другими дистрибутивами процесс

может затянуться. Для *Minecraft* должен быть запущен X-сервер, и если вы любитель консоли, наберите **startx**. Запустите *LXTerminal*, после чего распакуйте и запустите содержимое архива следующим образом:

```
$ tar -xvzf minecraft-pi-0.1.1.tar.gz
$ cd mcpi
$ ./minecraft-pi
```

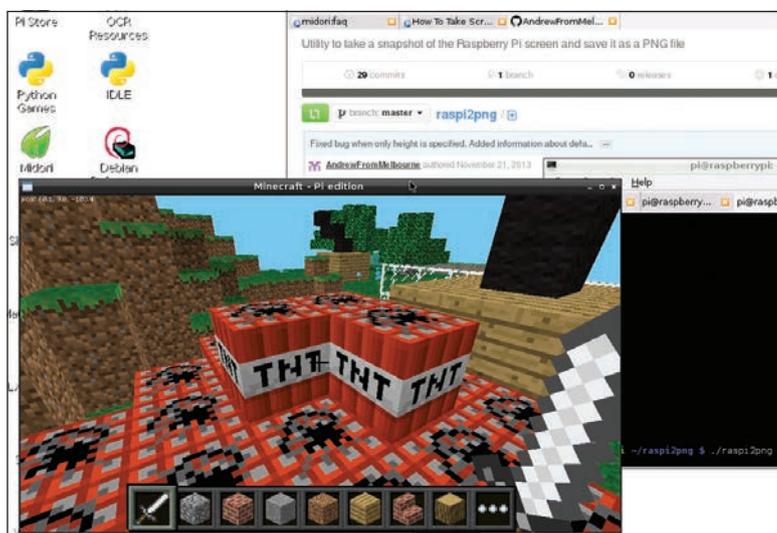
Правда ведь, как гладко все прошло? В левом верхнем углу вы видите координаты x, y и z, которые изменятся при переходе в блочную среду. Оси X и Z параллельны поверхности пола, а Y — это высота. Каждый блок (или, корректно выражаясь, воксель), из которых состоит пространство, описывается целочисленными координатами и типом блока [BlockType]. Пол не имеет глубины: вместо этого считается, что он состоит из плиток. Пустое пространство имеет тип блока AIR [воздух]; кроме него, есть еще около 90 более осозаемых веществ, включая такие прельстительные, как GLOWING_OBSIDIAN [светящееся вулканическое стекло] и TNT [взрывчатка]. У координат игрока, в отличие от координат блоков, есть десятичная часть, благодаря чему игрок может перемещаться внутри пустых блоков AIR.

API позволяет подключиться к работающему экземпляру *Minecraft* и управлять игроком и местностью в соответствии со своей манерой величия. С целью эту манеру убажить, первым делом скопируем предоставленную библиотеку, чтобы не портить чистую установку *Minecraft*. Для нашей личной вотчины создадим специальный каталог `~/picraft` и поместим все API-файлы в `~/picraft/minecraft`. Откройте *LXTerminal* и скомандуйте:

```
$ mkdir ~/picraft
$ cp -r ~/mcpi/api/python/mcpi ~/picraft/minecraft
```

Теперь без долгих проволочек перейдем к первым реформам в *Minecraft*. Начнем с запуска интерактивного сеанса Python в *Minecraft*; для этого откройте еще одну вкладку в *LXTerminal*, запустите *Minecraft* и войдите в мир, затем с помощью Alt+Tab вернитесь в терминал и откройте в другой вкладке Python. На вкладке с Python сделайте вот что:

```
import minecraft.minecraft as minecraft
```



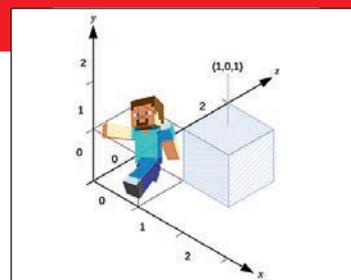
➤ Не делайте этого дома, детки... а вообще-то — *сделайте* это дома.

Эй, а где мой Стив?

Наш бесстрашный герой (звать Стив) находится в блоке (0,0,0). Он может перемещаться внутри этого блока, а несколько шагов в направлении осей X и Z приведут Стива в блок, затененный голубым. Во время этого короткого путешествия он будет находиться сразу в двух блоках, но функция `getTilePos()` API *Minecraft* выберет тот блок, в котором находится большая его часть.

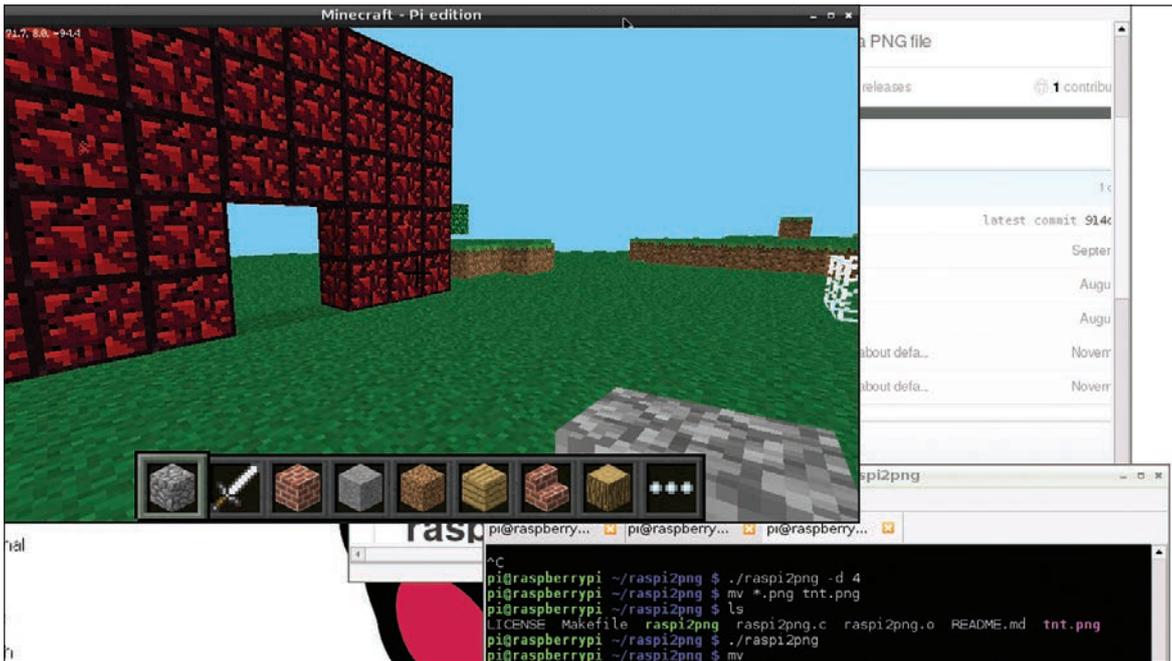
При попытке преобразовать стандартные понятия, такие как линии и многоугольники, из Евклидова пространства в дискретные блоки возникают некоторые

тонкости. Двумерная версия этой проблемы возникает каждый раз при рендеринге любой векторной графики. Предположим, вам нужно нарисовать линию между двумя точками на экране. В этом случае, если линия не горизонтальная и не вертикальная, нужно выбирать, какие пиксели закрасить. Самое раннее решение этой проблемы предложил Джек Элтон Брезенхем [Jack Elton Bresenham] в 1965 году, и мы расширим этот классический алгоритм на трехмерное пространство в следующей статье (см. Учебники, стр. 88).



➤ Мир *Minecraft* красуется здесь благодаря изометрической проекции.

» Все самые немислимые конструкции могут быть вашими.



```
import minecraft.block as block
mc = minecraft.Minecraft.create()
posVec = mc.player.getTilePos()
x = posVec.x
y = posVec.y
z = posVec.z
mc.postToChat(str(x)+' '+str(y)+' '+str(z))
```

Ага! Точка, где мы находимся, на пару секунд засветилась (если нет, вы в чем-то ошиблись). Эти координаты относятся к текущему блоку, где окопался персонаж, и поэтому у них нет десятичной части. Сравнив их с координатами в левом верхнем углу, вы поймете, что они — просто результат округления этих вещественных координат к целым (например, -1.1 округлена до -2). Координаты вашего героя можно получить из функции **mc.player.getPos()**, так что **getTilePos()** отчасти избыточна, но в ней три вещественные координаты также преобразуются к целым, и можно воспользоваться и ею. В API есть отличный класс **Vec3** для работы с трехмерными векторами, такими как положение нашего героя. Он включает все стандартные векторные операции, типа сложения и скалярного произведения, а также более экзотические штуки, которые помогут нам в дальнейшем.

```
Также можно получить данные, на чем стоит наш персонаж.
Вернитесь в сеанс Python и наберите команды
curBlock = mc.getBlock(x, y - 1, z)
mc.postToChat(curBlock)
```

Здесь, **getBlock()** возвращает целое число, определяющее тип блока: 0 означает воздух, 1 — камень, 2 — траву, а все остальные типы можно найти в файле **block.py** в ранее созданном каталоге **~/picraft/minecraft**. Мы вычитаем единицу из координаты у, так как хотим посмотреть, что происходит у нас под ногами — вызов функции **getBlock()** в текущем местоположении должен всегда возвращать 0, так как в противном случае наш персонаж был бы вмурован во что-нибудь твердое или тонул бы в чем-нибудь жидком.

Как обычно, интерпретатор Python — отличное место для экспериментов, но зрелый подход — помещать свой код в файл. Создайте файл **~/picraft/gps.py** со следующим кодом.

```
import minecraft.minecraft as minecraft
import minecraft.block as block
mc = minecraft.Minecraft.create()
oldPos = minecraft.Vec3()
```

```
while True:
    playerTilePos = mc.player.getTilePos()
    if playerTilePos != oldPos:
        oldPos = playerTilePos
        x = playerTilePos.x
        y = playerTilePos.y
        z = playerTilePos.z
        t = mc.getBlock(x, y - 1, z)
        mc.postToChat(str(x) + ' ' + str(y) + ' ' + str(z) + ' ' + str(t))
    Теперь запустите Minecraft, войдите в мир, затем откройте терминал и запустите свою программу:
    $ python gps.py
```

В работающей программе с каждым перемещением героя должны отображаться тип блока, на котором он стоит, и его координаты. Когда запомните все типы блоков (шутка), нажмите **Ctrl+C** для выхода из программы.

Мы рассказали о некоторых «пассивных» опциях API, но они интересны только в сочетании с более конструктивными (или деструктивными) опциями. Прежде чем закончить, расскажем о некоторых из них. Как и прежде, запустите **Minecraft** и откройте сеанс Python, импортируйте модули **Minecraft** и блоков и создайте объект **mc**:

```
posVec = mc.player.getTilePos()
x = posVec.x
y = posVec.y
z = posVec.z
for j in range(5):
    for k in range(x - 5, x + 5)
        mc.setBlock(k, j, z + 1, 246)
```

Смотрите! Рядом с нашим персонажем выросла стена размером 10×5 из раскаленного обсидиана. Можно удалять блоки, превращая их в воздух — например, проделать в нашей стене небольшой тоннель (при этом, разумеется, не стоит перемещать персонажа после ввода предыдущего кода):

```
mc.setBlock(x, y, z + 1, 0)
```

В следующий раз мы посмотрим, как создавать и разрушать более сложные конструкции, займемся физикой, перепишем некоторые ее законы и вообще вволю надуркуемся в пределах нашего мира 256×256×256. А пока — почему бы не пошалить с функцией **mc.player.setPos()**? В конце концов, телепортация — это прелево. **LXF**

Скорая помощь

Загляните на сайт Мартина О'Хэнлона [Martin O'Hanlon] www.stuffaboutcode.com: там есть несколько прекрасных примеров того, на что способен API.

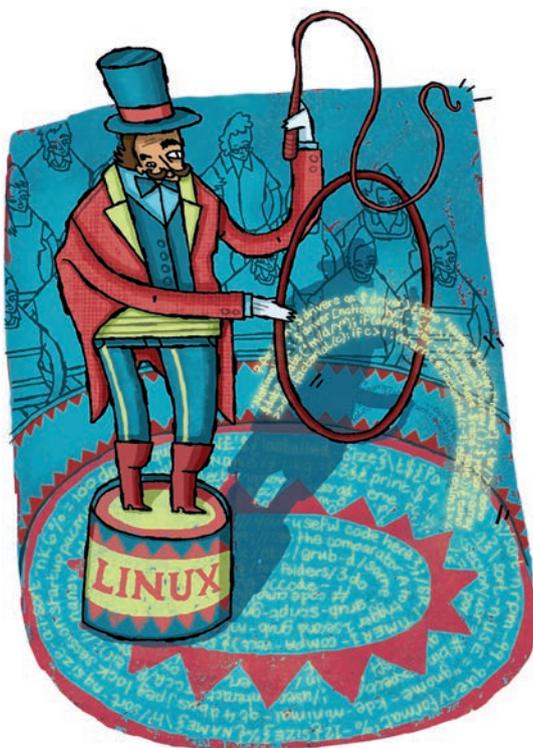
Minecraft: Требушет на Pi

Любовно выстроим дом, а потом пустим его к небесам с помощью пиротехника Джонни Бидвелла, бруска взрывчатки и средневековой осадной машины.



Наш
эксперт

Джонни Бидвеллу хотелось бы побольше блочных вещей в этой чертовой вселенной, зачем-то сделанной из кварков.



Итак, мы ознакомились с основами API, и пора сделать что-нибудь по-настоящему творческое. Построить дом трудно, не так ли? А вот и не так. Всего несколько строк на прекрасном Python — и дом вашей мечты готов. По крайней мере, если дом вашей мечты — стандартная «коробка»: чем безудержнее мечта, тем больше потребуются кода. Вам не придется беспокоиться об охране, подводе коммуникаций, оплате за регистрацию недвижимости, и вы никогда не наткнетесь на могильник эпохи неолита (если только сами предварительно не создали его).

В *Minecraft Pi* не бывает дождей, и нам прекрасно подойдет плоская крыша. Для начала создадим два угла нашего дома: $v1$ — ближайший к нам блок, расположенный по оси X и на один блок выше по оси Y, и $v2$ — на подходящем расстоянии от первого блока:

```
pos = mc.player.getTilePos()
v1 = minecraft.Vec3(1,1,0) + pos
v2 = v1 + minecraft.Vec3(10,4,6)
```

Теперь создадим сплошной прямоугольный параллелепипед между этими вершинами и затем сделаем его полом, создав внутри еще один прямоугольный параллелепипед из свежего воздуха:

```
mc.setBlocks(v1.x,v1.y,v1.z,v2.x,v2.y,v2.z,4)
```

```
mc.setBlocks(v1.x+1,v1.y,v1.z+1,v2.x-1,v2.y,v2.z-1,0)
```

Отлично; вот только входить и выходить нужно через непомерно большой проем в крыше, да и неплохо было бы сделать хороший деревянный пол (поняли фишку?). Если вы стоите на плоской поверхности, то заметите, что стены нашего дома подняты на один блок над поверхностью земли. В этом пространстве мы поместим наш пол. Если ваша местность не такая ровная, то дом может стоять на холме или частично висеть в воздухе, но не волнуйтесь: мы это исправим, сформировав нужный рельеф или внося изменения в локальную гравитацию. Создадим наш грубый деревянный пол:

```
mc.setBlocks(v1.x,v1.y-1,v1.z,v2.x,v1.y-1,v2.z,5)
```

По аналогии создадим окна:

```
mc.setBlocks(v1.x,v1.y+1,v1.z+1,v1.x,v1.y+2,v1.z+3,102)
```

```
mc.setBlocks(v1.x+6,v1.y+1,v1.z,v1.x+8,v1.y+2,v1.z,102)
```

```
mc.setBlocks(v2.x,v1.y+1,v1.z+1,v2.x,v1.y+2,v1.z+3,102)
```

```
mc.setBlocks(v1.x+2,v1.y+1,v2.z,v1.x+4,v1.y+2,v2.z,102)
```

Для крыши используется специальный полублок 44, который может быть различных типов. Мы сделаем крышу деревянной, как и пол, задав тип блока:

```
mc.setBlocks(v1.x,v2.y,v1.z,v2.x,v2.y,v2.z,44,2)
```

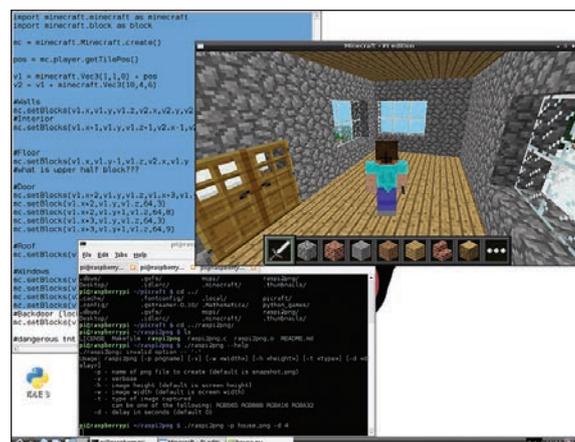
С дверью все чуть сложнее (душераздирающие подробности описаны во врезке на стр. 90), но для нее нужны следующие три строки кода:

```
mc.setBlocks(v1.x+2,v1.y,v1.z,v1.x+3,v1.y,v1.z,64,3)
```

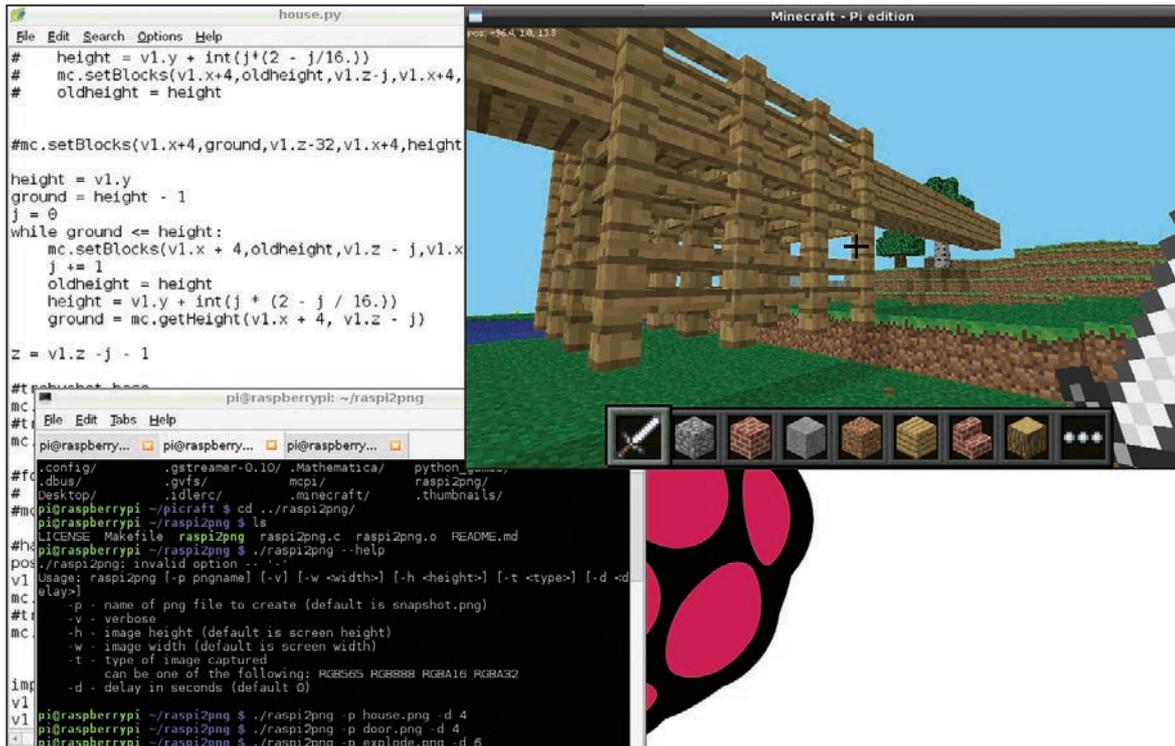
```
mc.setBlock(v1.x+2,v1.y+1,v1.z,64,8)
```

```
mc.setBlock(v1.x+3,v1.y+1,v1.z,64,9)
```

С любовью и усердием построив новый дом, приступим к поиску новых и изобретательных способов разрушить его. Мы уже упомянули о взрывчатке, и элегантным ударом меча (или еще



» Дом наш; захотим — и взорвем его.



» На вид негусто... но еще не вечер.

чего-нибудь) ее можно взорвать. Заполнить дом блоками взрывчатки слишком тривиально: есть способы получше. Читатель, пожалуйста, представьте вам бета-версию моего требушета.

Вместо полета снаряда по параболической траектории в космическом пространстве мы сделаем так, чтобы по этой самой параболе несло взрывчатку. Взрыв в исходной точке вызовет отменную цепную реакцию, кульминацией которой станет разрушение львиной доли вашего дома. Для начала расскажем о некоторых основах двумерной механики. В отсутствие сил трения снаряд описывает параболу, определяемую первоначальной скоростью запуска, углом запуска и ускорением свободного падения, которое на Земле составляет около 9,81 м/с².

В порядке изящного введения мы зададим эти константы так, чтобы горизонтальное расстояние, покрываемое этой дугой, составляло ровно 32 блока, а максимальная высота — 16 блоков по отношению к исходной высоте. Если длина каждого блока 1 м, то начальная скорость для нашего примера составила бы скромные 18 м/с, а угол запуска — 60 градусов. Мы ограничимся двумя измерениями, поэтому дуга будет проходить по оси Z, а координата по X будет постоянной — рядом с дверью. Все это сводится к простой формуле $y = z(2 - z/16)$, которую мы реализуем так:

```
for j in range(33):
    height = v1.y + int(j*(2 - j/16.))
    mc.setBlock(v1.x+4,height,v1.z-j,46,1)
```

Последний параметр запалит взрывчатку — рубаните мечом и наслаждайтесь фейерверком. А возможно, все будет иначе: в результате взрывов (которые, кстати, сильно грузят «Мозги» Pi) часть взрывчатки упадет, прервав цепочку, и наш любимый дом уцелеет. Нам этого не надо, так что воспользуемся следующим кодом:

```
height = v1.y
ground = height - 1
j = 0
while ground <= height:
    mc.setBlocks(v1.x + 4,oldheight,v1.z - j,v1.x + 4,height,v1.z - j,46,1)
```

```
j += 1
oldheight = height
height = v1.y + int(j * (2 - j / 16.))
ground = mc.getHeight(v1.x + 4, v1.z - j)
```

Тогда мы гарантируем, что парабола не будет содержать разрывов, а также то, что цепочка взрыва не прервется еще в воздухе. Последнюю проблему мы побороли так: с помощью функции `getHeight()` определяем уровень земли для каждой точки дуги и перестаем строить дугу при достижении земли. Обратите внимание, что функцию `getHeight()` нужно вызвать до размещения последнего блока взрывчатки, так как высота мира определяется самым верхним блоком, который не является воздухом, даже если и висит в воздухе.

Если наша конструкция выходит за пределы мира *Minecraft*, можно построить еще один дом в более удобном месте или изменить `v1.z` — `j` на `max(-116,v1.z-j)` в цикле выше; тогда у нас получится вертикальный тотем угрозы прямо на границе мира. Теперь траектория у нас задана, и можно добавить мощную осадную машину:

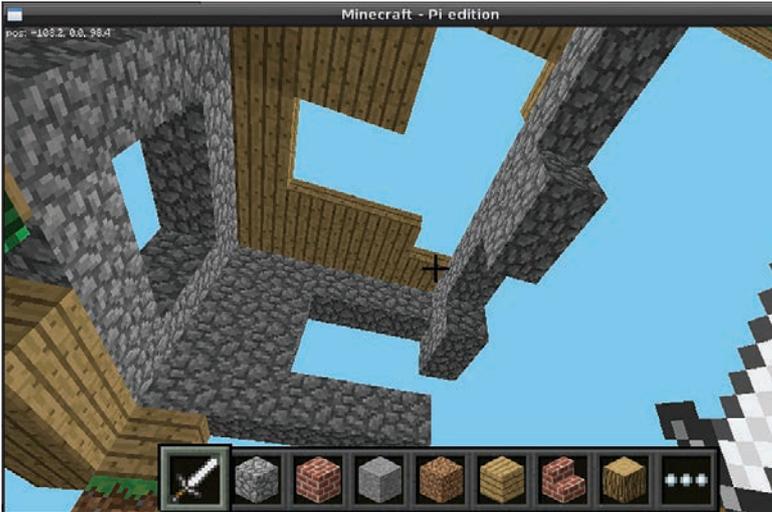
```
z = v1.z - j - 1
mc.setBlocks(v1.x + 3, oldheight, z + 10, v1.x + 6, oldheight + 2, z + 7,85)
mc.setBlocks(v1.x + 4, oldheight + 2, z + 12, v1.x + 4, oldheight + 2, z + 1, 5)
```

До этой точки мы выравнивали все по конкретным осям — наш дом (перед тем его как взорвали) смотрел по оси Z в отрицательном направлении, что эквивалентно направлению «на юг», и туда же смотрит наша парабола. Нашу сцену можно повернуть на 90 градусов, и код почти не изменится — не считая некоторых перестановок `x`, `y`, `z` и +/- — хотя дом, поставленный на стену, будет выглядеть несколько экзотически. Все усложнится, если мы захотим сбросить с себя узду этой сетки и правильных углов и поработать с произвольными углами. Проблема в том, как построить прямую линию, когда наши фундаментальные единицы — блоки с фиксированным положением, а не точки.

Скорая помощь

Код требушета я написал, вдохновленный удивительным Марином О'Хэнлоном и его проектами для Pi на сайте www.stuffaboutcode.com. Подробнее про Мартина — на стр. 40.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



➤ Дом здорово продырявили! (После нашего урока его придется чинить.)

В ваших последующих проектах бесценной окажется общая трехмерная функция **drawline()**, позволив вам создавать всякие конструкции от параллелепипедов до пентаграмм. Для нее нам понадобится 3D-версия классического алгоритма Брезенхама [Bresenham]. В Github гуру Pi Мартина О'Хэнлона есть несколько чудесных проектов для *Minecraft Pi Edition*, включая мощную пушку, которая вдохновила мой проект. У Мартина есть целый класс рисования на Python, который включает вышеупомянутый трехмерный алгоритм построения линии, но если вы понимаете двумерный алгоритм, обобщить его довольно просто.

Представим, что мы находимся в равнинном мире Minecraft и хотим рассчитать линию в плоскости (X,Y), соединяющую точки (-2,-2) и (4,1). Уравнение этой линии — $y = 0.5x - 1$. По требованию алгоритма, градиент линии должен быть от 0 до 1, и здесь все отлично. Будь нам нужна линия с другим наклоном, мы поменяли бы местами оси так, чтобы градиент входил в нужный диапазон. Основная идея алгоритма состоит в том, что в строке пикселей заполняется только по одному пикселю (блоку) в каждом столбце, но по несколько в каждой строке. Поэтому в направлении X мы будем закрасивать пиксель за пикселем, а координата по Y будет оставаться постоянной или увеличиваться на единицу. Простой код на Python, который это реализует, будет таким:

```
dx = x1 - x0
dy = y1 - y0
y = y0
```

```
error = 0
grad = dy/dx
for x in (x0,x1):
    plot(x,y)
    error = error + grad
    if error >= 0.5:
        y += 1
    error -= 1
```

Здесь **plot()** — воображаемая функция построения графика, а **grad** принимает значения между 0 и 1. Таким образом, мы увеличиваем **y** на единицу каждый раз при накоплении достаточной ошибки, и результат радует глаз.

Идея Брезенхама состояла в том, чтобы свести все вычисления к целочисленным операциям, приносившаяся к производительности «железа» 1960-х. Сегодня нам доступны вычисления с плавающей точкой с огромной скоростью, но все равно стоит отдать должное этим новаторским приемам. Переменные с плавающей точкой **grad** и **error** увеличиваются за счет деления на **dx**, и если умножить все на это количество и обойти масштабирование, все будет отлично.

Чтобы это работало в трех измерениях, не требуется большого усилия мысли: сначала нужно определить доминантные оси (оси с самым большим изменением координаты), а затем соответственно повернуть оси. После чего перемещаться по доминантным осям по одному блоку и соответственно увеличивать координаты младших осей. Нужно внимательно следить за знаком изменения каждой координаты, которое мы храним в переменной **ds**. Функция **ZSGN()** возвращает 1, -1 или 0, если ее аргумент положительный, отрицательный или равен нулю соответственно; напишите эту функцию сами в качестве упражнения. Мы усиленно применяем вспомогательную функцию **minorList(a,j)**, которая возвращает копию списка после удаления j-го элемента. Благодаря функции **lambda** и разрезанию списка ее можно записать одной строкой:

```
minorList = lambda a,j: a[:j] + a[j + 1:]
```

Функция **getLine()** берет две вершины, каждая из которых представлена списком из трех координат, и возвращает список всех вершин в получившейся трехмерной линии. Вся эта функция основана на коде Мартина, за который мы должны сказать ему спасибо. В первой части кода инициализируется наш массив вершин и обрабатывается простой случай, в котором начальная и конечная вершины совпадают...

В этом случае наша линия представляет собой всего один блок:

```
def getLine(v1, v2):
```

Скорая помощь

Весь код можно написать прямо в интерпретаторе, но ошибки при его копировании будут раздражать. Поэтому, возможно, проще поместить код в файл **house.py** и запускать его командой **python house.py** при работе с *Minecraft*.

Как построить двойную дверь

Нашему дому нужны двери, и здесь мы впервые встречаем дополнительный параметр **blockData**. Это целое число от 0 до 15, и оно управляет добавочными свойствами блоков, такими как цвет дерева или тем, подожгли мы взрывчатку или нет. Наша дверь занимает четыре блока и выровнена по оси X. Она немного углублена в окружающие стены, закрыта, а посреди каждой двери расположены удобные ручки. На все эти свойства влияют различные биты **blockType**. Мы нумеруем четыре бита с крайнего справа до крайнего слева от 0 до 3 и используем прямой порядок байтов, т.е. двоичным представлением числа 8 будет 1000. Бит 3 устанавливается в 1, если блок входит в верхнюю часть

двери. Если это так, то единственный другой бит, который нас интересует — бит 0, он определяет положение ручек/петель. Верхняя часть двери состоит из блоков типа 8 или 9.

Для нижних частей двери используются следующие значения битов:

- бит 3 установлен в 0
- бит 2 дверь открыта
- бит 1 дверь закрыта
- бит 0 выравнивание (0 — по оси X, 1 — по оси Z)

Верхние части двери следует размещать во вторую очередь, после нижних, поскольку они наследуют свойства нижних частей.



➤ Двери всегда нужны тем, кто хочет избежать клаустрофобии/кончины.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

```
if v1 == v2:
```

```
    vertices.append([v1])
```

После этого наша функция становится немного уродливой. Мы задаем вышеупомянутый список знаков **ds** и список абсолютных разностей (умноженных на два) **a**. Строка **idx =** с технической точки зрения не лучший вариант, мы хотим определить доминирующие оси, то есть индекс максимального элемента в **a**. Использование метода **index()** совместно с **max** означает, что мы проходим по нашему списку дважды; но поскольку список очень короткий, беспокоиться не о чем, зато код выглядит гораздо лучше. Доминантные координаты мы обозначаем как **X** и **X2**. Наш список **s** представляет собой переупорядоченный **ds** с доминантными координатами в начале. Также есть некоторые другие списки для отслеживания ошибок. Переменная **aX** содержит знак изменения координат по доминантным осям:

```
else:
```

```
    ds = [ZSGN(v2[j] - v1[j]) for j in range(3)]
```

```
    a = [abs(v2[j]-v1[j]) << 1 for j in range(3)]
```

```
    idx = a.index(max(a))
```

```
    X = v1[idx]
```

```
    X2 = v2[idx]
```

```
    delta = a[idx] >> 1
```

```
    s = [ds[idx]] + minorList(ds,idx)
```

```
    minor = minorList(v1,idx)
```

```
    aminor = minorList(a,idx)
```

```
    dminor = [j - delta for j in aminor]
```

```
    aX = a[idx]
```

Задав все эти переменные, можно углубиться в главный цикл, где добавляются вершины, анализируются разности по младшим осям, повторно вычисляются ошибки и увеличиваются основные координаты. Функция возвращает желанный список вершин.

```
loop = True
```

```
while(loop):
```

```
    vertices.append(minor[:idx] + [X] + minor[idx:])
```

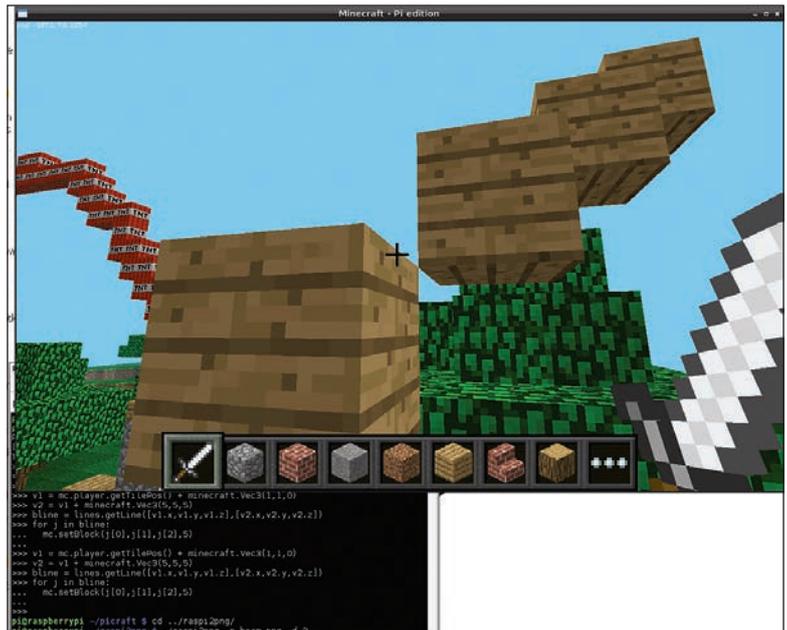
```
    if X == X2:
```

```
        loop = False
```

```
    for j in range(2):
```

```
        if dminor[j] >= 0:
```

```
            minor[j] += s[j + 1]
```



```
dminor[j] -= aX
```

```
dminor[j] += aminor[j]
```

```
X += s[0]
```

```
return vertices
```

Чтобы закончить стильно, протестируем эту функцию, создав загадочную и шаткую деревянную балку рядом с тем местом, где мы стоим, как иллюстрацию нашего вдохновенного труда. Ай да мы!

```
v1 = mc.player.getTilePos() + minecraft.Vec3(1,1,0)
```

```
v1 = minecraft.Vec3(1,1,0) + pos
```

```
v2 = v1 + minecraft.Vec3(5,5,5)
```

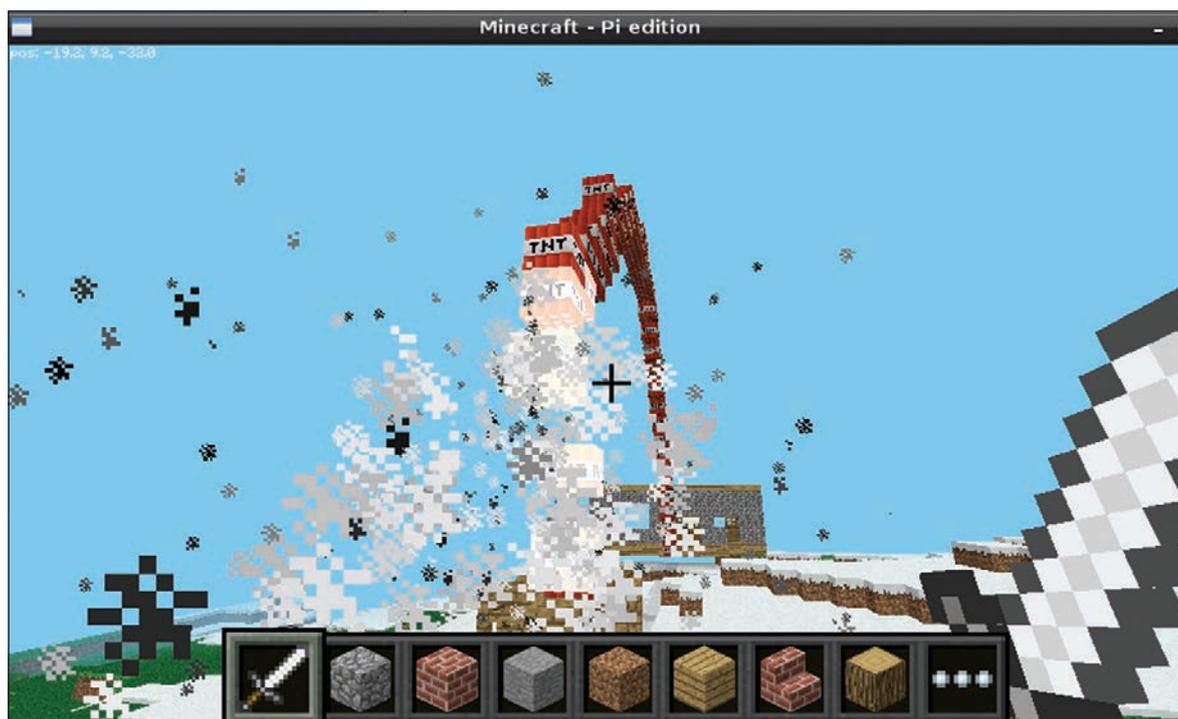
```
bline = getLine([v1.x,v1.y,v1.z],[v2.x,v2.y,v2.z])
```

```
for j in bline:
```

```
    mc.setBlock(j[0],j[1],j[2],5)
```

На следующем уроке мы создадим в *Minecraft* полноценную пушку. Пли! **LXF**

➤ Теперь мы можем забыть про сетку и выбирать любые углы, какие пожелаем.



➤ Не делайте этого дома, детки.

MySQL: Работа с электронными таблицами

Запросто преобразуем таблицы в SQL-запросы и обратно с *MySQL* и *phpMyAdmin*. Канада нам поможет — в лице **Кента Ельчука**.



Наш эксперт

Кент Ельчук — ведущий web-разработчик в двух колледжах Ванкувера. Он программирует, пишет книги и ведет блог, заодно экспериментируя с виртуальными машинами.

MySQL — самая популярная открытая СУБД в мире, и она знакома каждому разработчику динамических web-сайтов или приложений, а также администратору удаленных баз данных.

MySQL бесплатна и работает в Linux и Windows. Ее можно установить за несколько минут и тут же приступить к работе. Хотя в динамических сайтах для взаимодействия с *MySQL* обычно используются языки программирования серверной стороны, типа PHP или Perl, вам очень часто придется взаимодействовать с ней напрямую — особенно при выполнении запросов к БД, для поиска и для вывода данных в файлы CSV или электронные таблицы.

Если вы новичок в *MySQL* и базах данных в целом, вкратце расскажу, как ими пользоваться. У базы данных есть уникальное имя, и она содержит различные таблицы с данными, в каждой из которых есть разнообразные столбцы, где хранятся данные. В *MySQL* вы будете постоянно сталкиваться с таблицами и столбцами, и если вы хотите облегчить себе работу, важно хорошо понимать, что это такое.

За время существования *MySQL* ряд разработчиков написали несколько клиентских утилит для администрирования баз данных *MySQL*. Одна из популярных утилит, которая работает через веб-интерфейс — *phpMyAdmin*. Это очень проворная и полезная утилита, позволяющая справляться со многими задачами, касающимися баз данных.

Однако управлять базами данных можно и с консоли *MySQL*. На самом деле, выбор той или иной утилиты — дело вкуса пользователя. Некоторые пользуются и *MySQL*, и *phpMyAdmin*,

выбирая утилиту в зависимости от конкретной задачи. На этом уроке мы расскажем о различных идеях и процедурах, используемых при работе в *phpMyAdmin* и консоли *MySQL*. В отличие от *phpMyAdmin*, которая работает в браузере, консоль *MySQL* запускается простой командой Linux, которая называется **mysql**.

Если в *MySQL* не задан пароль, достаточно набрать эту команду и нажать Enter, чтобы открылась командная строка, которая выглядит так: **mysql>**.

<input type="checkbox"/>	9	blank_4	in
<input type="checkbox"/>	10	fires	in
<input type="checkbox"/>	11	site_completed	va
<input type="checkbox"/>	12	extinguished_date	da
<input type="checkbox"/>	13	blank_8	in
<input type="checkbox"/>	14	blank_9	in
<input type="checkbox"/>	15	blank_10	in

► При выборке данных из таблиц по ключам каждый ключ должен быть индексируем или быть первичным ключом. Тогда запросы будут выполняться быстрее.

Шпаргалка по операторам MySQL

Вот краткое описание основных операторов *MySQL*:

- » **LIKE** ищет точное соответствие. Очень похож на =.
- » **%LIKE%** ищет любое вхождение в строке, будь то полное или частичное соответствие содержимому ячейки.
- » **NOT LIKE** ищет данные, которые не соответствуют введенному тексту.
- » = используется для поиска точного соответствия.
- » **!** находит данные, которые не соответствуют указанной строке.

Операторы регулярных выражений пригодятся тем, кто знает, как составлять регулярные выраже-

ния с такими шаблонами, как REGEXP, REGEXP ^...\$, NOT REGEXP. Это отдельный мир, и в нем обитают более опытные пользователи.

- » **=>** ищет пустые строки.
- » **!=>** ищет непустые строки.
- » **IN()** — оператор, который проверяет наличие в столбцах элементов с одним или несколькими значениями. Значения разделяются запятыми. Например, **IN(3,6)** вернет все строки, у которых в данном столбце содержится значение либо 3, либо 6.
- » **NOT IN()** — противоположность **IN()**.

» **BETWEEN** — оператор, используемый для поиска чисел и дат в заданном диапазоне. Например, если нужно найти данные в диапазоне дат, можно воспользоваться конструкцией **BETWEEN <2014-04-01> AND <2014-04-30>**.

» **NOT BETWEEN** — противоположность **BETWEEN()**.

» **IS NULL** означает, что значение отсутствует или неизвестно. Одни программисты предпочитают заносить во все пустые ячейки NULL, другие — пустые строки.

» **IS NOT NULL** означает, что значение поля определено.

Если пароль задан, нужно набрать команду `mysql -u root -p` и нажать Enter. После этого просто наберите пароль и снова нажмите Enter. У вас появится доступ к консоли *MySQL*.

Если вы хотите вставить команды *MySQL* в *phpMyAdmin*, это делается так же, как и с любым текстовым редактором, браузером и т.п. А вот для вставки текста в консоль *MySQL* нужно нажать Shift+Insert после копирования.

Наконец, все запросы в *phpMyAdmin* можно завершать традиционной в *MySQL* точкой с запятой (;), а можно и не завершать. Команды в консоли *MySQL* завершать нужно обязательно.

Теперь, вкратце ознакомившись с *MySQL*, поговорим о том, как ею пользоваться, и посмотрим, как работать с электронными таблицами и файлами SQL. Это повседневные задачи для всех, кто регулярно работает с *MySQL*, и поэтому важно их понимать.

Загрузка документов

Если сохранить электронную таблицу в файл CSV (comma-separated values — значения, разделенные запятыми), импортировать его в базу данных *MySQL* будет легко. Эти данные можно поместить в пустую базу данных или в базу данных, где данные уже есть. Ограничения на такой перенос данных зависят от параметров БД. Например, если в ней запрещены дубликаты в одном или нескольких столбцах, то результаты будут различаться по сравнению с таблицей, в которую можно вставлять любые данные.

Главное при переносе данных — убедиться, что количество полей в таблице CSV и таблице базы данных совпадает. Например, если в электронной таблице пять столбцов, то и в таблице БД должно быть пять столбцов. Подготовив файл CSV и открыв *phpMyAdmin*, можно импортировать электронную таблицу, открыв таблицу БД, выбрав Import [Импорт], затем выбрав CSV в качестве формата [Format], нажав Browse [Обзор] и выбрав файл CSV. Если в вашей электронной таблице есть заголовки, поставьте галочку The First Line of the File Contains the Table Column Names [Первая строка файла содержит заголовки столбцов таблицы]. Если эту галочку не поставить, заголовки столбцов попадут в данные.

Желая импортировать файл CSV в новую таблицу, выполните те же действия, однако при одновременном заполнении и создании таблицы новая таблица получит название вроде "Table 2", а ее столбцы будут иметь названия "COL 1", "COL 2", "COL 3" и т.д.

Безликие имена обычно усложняют работу с таблицами и поэтому не подходят администраторам БД. Это особенно актуально с ростом количества таблиц, столбцов и индексов.

Желая переименовать таблицу, откройте ее, выберите Operations [Операции], измените название в текстовом поле и нажмите Go [Выполнить]. Точно так же можно переименовать базу данных. Для переименования столбца просто выберите Structure > Change [Структура > Изменить] для соответствующего столбца. Наберите новое имя и нажмите Save [Сохранить].

Осмысленные имена баз данных, таблиц и столбцов существенно упрощают работу с БД, сохраняя массу драгоценного времени.

Перенос данных в MySQL

На консоли *MySQL* можно импортировать данные в таблицу. В некоторых установках *MySQL* по умолчанию нужно изменить файл `/etc/mysql/my.cnf` и добавить строку `local-infile` под `mysqld` и `mysql`. Для импорта данных нужно открыть (использовать) именно ту базу данных и таблицу, куда должны попасть данные. Для импорта таблицы БД из файла CSV делается следующее:

```
LOAD DATA LOCAL INFILE 'colours_clothes.CSV' INTO TABLE
colours_clothes FIELDS TERMINATED BY ',' enclosed by '<<' 'lines
terminated by '\n' (column_1, column_2, column_3)
```

Если вы хотите импортировать таблицу БД из текстового файла, то здесь используется формат полей, разделенных табуляцией, и большие таблицы загружаются с впечатляющей скоростью. Например, большую таблицу со всеми почтовыми индексами, кодами городов и стран можно добавить в БД быстрее, чем вы вскопаете со стула. Перенос этой же таблицы из файла CSV или SQL потребовал бы гораздо больше времени. Вот пример списка одежды разных цветов, поля в котором разделены табуляцией. Файл называется `colours_clothes.txt`:

```
id colour type
1 red shirt
2 yellow hat
3 blue jeans
```

Он импортируется следующей командой:

```
LOAD DATA LOCAL INFILE '/home/username/colours_clothes.txt'
INTO TABLE colours_clothes FIELDS TERMINATED BY '\t' LINES
TERMINATED BY '\r\n' IGNORE 1 LINES;
```

Вот что делает команда. Текстовый файл загружается в таблицу `colours_clothes`. Поля завершаются символом `\t`, что означает, что столбцы разделены табуляцией. После каждого слова на каждой строке пробелы. `LINES TERMINATED BY '\r\n'` означает, что каждая новая строка файла — это новая строка таблицы. `IGNORE 1 LINES` означает, что первая строка не должна загружаться в таблицу — в ней лишь содержится описание каждого столбца.

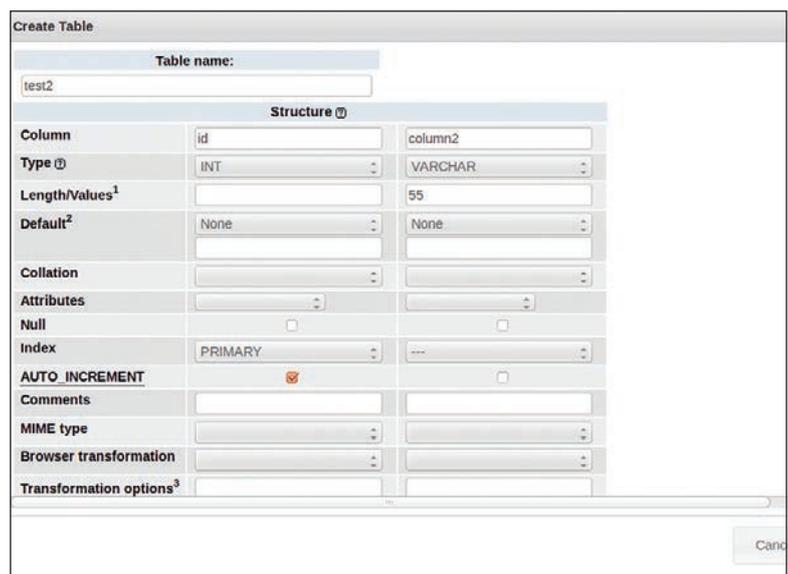
Поиск через PhpMyAdmin

Теперь займемся методами получения данных, которые должны появиться в вашем файле CSV. Мы также рассмотрим, как экспортировать данные в электронную таблицу, пригодную для открытия в *OpenOffice Calc*. Обычно требуемые данные можно получить с помощью поиска и SQL-запросов.

С помощью поиска можно получить практически любые нужные вам данные из одного или нескольких столбцов таблицы, а на вкладке SQL можно написать собственные запросы *MySQL* для получения данных из одной или нескольких таблиц. Поиск используется для получения данных широкого диапазона из одной, нескольких или всех баз данных или для получения необходимых данных из одного или нескольких столбцов таблицы базы данных.

Скорая помощь 

Через поиск в *phpMyAdmin* данные найдутся быстро. Но настройте запрос в консоли *MySQL* или на вкладке SQL в *phpMyAdmin*, вы получите конкретные данные из различных таблиц.



» При создании таблицы для каждого столбца задается имя, тип, сопоставление (при необходимости) и добавляется индекс (также при необходимости).

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Для поиска по всем таблицам откройте phpMyAdmin и выберите необходимую БД, затем перейдите на вкладку Search [Поиск]. В поле в верхней части вкладки введите слово или строку, которую вы ищете. После этого выберите все или некоторые таблицы в качестве области поиска. Для выбора отдельных таблиц щелкайте по ним мышью, удерживая клавишу Ctrl. Выбрав таблицы, нажмите Go.

Поиск по базе данных

Чтобы сузить область поиска до одной таблицы, выберите базу данных и таблицу и перейдите на вкладку Search. Откроется список всех столбцов таблицы с указанием названия, типа, сопоставления, оператора и значения для каждого столбца. Для поиска нам прежде всего понадобятся оператор и значение.

Операторами можно задавать различные условия для значения, которое вы ищете. Например, если вы хотите найти имя в столбце с именами, можете указать значение "Peter". Если после этого нажать Go, программа будет искать значение "Peter" во всех строках и выведет строки, в которых в этом столбце указано слово "Peter". По умолчанию для большинства типов **varchar** и других типов столбцов оператором поиска по умолчанию служит LIKE, а для большинства целочисленных типов — знак равенства (=).

Название оператора LIKE может ввести в заблуждение, потому что на самом деле он ищет точное соответствие. Однако если его немного видоизменить (%LIKE%), то в результаты поиска попадут и любые вхождения в строку. Например, если указать после оператора %LIKE % слово "Peter", то в результат попадут строки со значениями "Peter", "Peter Jones" и "Dave Peter".

Написание запросов *MySQL* требует известной практики. При сборе данных из нескольких таблиц нужно убедиться, что

у разных таблиц есть как минимум одно совпадение по значению с другой таблицей. Такие совпадения называются ключами.

Например, если у вас есть таблица участников с идентификатором участника member_id, то мы сможем получить записи в блоге для этого участника, если в таблице blog_entries будет столбец, содержащий те же идентификаторы участников member_id. В свою очередь, в таблице blog_entries может быть столбец blog_id, который соответствует столбцу blog_id в таблице blog_comments. Так мы свяжем таблицу blog_comments с таблицей blog_entries.

Если таблицы связаны совпадающими индексами (ключами), можно получить любые нужные вам данные. Хотя можно связать таблицы с помощью соответствующих ключей (значений в столбце), можно также создать для столбца индексы. При получении данных из нескольких таблиц индексы и первичные ключи позволяют значительно ускорить выполнение запроса.

Написание запросов MySQL

Рассмотрим несколько простейших запросов. Вот запрос на получение всех строк из одной таблицы:

```
SELECT * FROM tablename
```

Следующий запрос выбирает только имена [name] из всех строк одной таблицы:

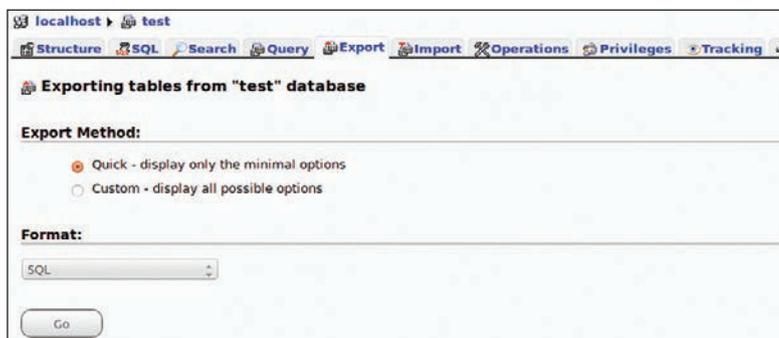
```
SELECT name FROM tablename
```

Продвинемся дальше. Ниже приведен пример с соединением таблиц реляционной базы данных. В отличие от запросов, приведенных выше, в этом запросе задаются условия с помощью WHERE. Запрос получает из различных таблиц именно те данные, которые нужны. Слово **pay** — алиас для таблицы payroll, а **slo** — алиас для таблицы site_list_official.

```
SELECT DISTINCT pay.site, slo.lat_dd, slo.long_dd, slo.lat_dmd, slo.long_dmd, pay.extinguished_date, (SELECT SUM(fires) FROM payroll pr WHERE pr.site = pay.site) AS fires FROM payroll as pay, site_list_official as slo WHERE pay.site = slo.site_number AND pay.extinguished_date = <2014-03-11>
```

Для добавления индекса в phpMyAdmin откройте таблицу и выберите Structure. Щелкните по выпадающему списку More [Далее] и нажмите Add Index [Добавить индекс].

Если вы планируете использовать базу данных с несколькими таблицами, то гораздо удобнее, когда в таблицах и базах данных используется одинаковое сопоставление, и все столбцы, которые используются при объединении, также имеют одинаковое сопоставление друг с другом. Сопоставление для таблицы легко изменить в phpMyAdmin. Откройте таблицу, выберите Change у нужного столбца, затем выберите Collation [Сопоставление] из выпадающего списка и нажмите Save. Важно помнить, что



➤ Всего парой щелчков мыши можно создать частичную и полную копию необходимых данных.

Советы и хитрости

Получив файл CSV с данными для импорта в таблицу базы данных, вы можете обнаружить, что этот файл пришел из источника, данные в котором постоянно меняются. В этом случае можно добавить в таблицу все строки, а затем позднее удалить все строки и добавить новые данные. Если вы работаете с постоянно меняющейся таблицей или с файлом CSV, то можно оказаться в ситуации, когда нужно быстро удалить все строки и добавить новые данные. С этим безмерно поможет команда **truncate**. Она очень проста: `truncate tablename;`

Вам могут пригодиться и хранимые процедуры. Они позволяют задавать переменные и выполнять запросы, запомненные и вызываемые при необходимости. Хранимые процедуры помогут, если у вас есть какие-то специализированные запросы, которые вы хотите сохранить и обращаться к ним при необходимости. Вот пример простой хранимой процедуры:

```
DELIMITER //
CREATE PROCEDURE my_new_procedure()
BEGIN
```

```
DECLARE my_variable INT DEFAULT 0;
SET my_new_variable = 15;
select my__new_variable;
END //
DELIMITER ;
```

После создания хранимой процедуры можно обратиться к ней из скрипта на PHP, Python или Perl на стороне сервера. Следующие команды показывают, как получить данные из хранимой процедуры:

```
$command = «CALL my_new_procedure()»;
$result = mysql_query($command, $db);
```

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

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 Извлечение VOB-файлов
- 2 Клавиатуры, которые не отвечают
- 3 Неконтролируемый возврат каретки в скриптах
- 4 Настройку твердотельных жестких дисков
- 5 Настройку MIME-типов в Firefox
- 6 Исправление ошибок с клавиатурой и мышью в Ubuntu 13.10

1 Перенос видео на DVD

В Я пытаюсь выудить короткий ролик с DVD, где таких роликов несколько, и записать его на другой диск. Диск некоммерческий и, насколько я знаю, не защищен. Я воспользовался учебником из LXF157, доступным в онлайн-архиве журнала — и все отлично получилось. С помощью *Handbrake* я преобразовал свой видеофайл в формат MP4, который воспроизводится в *VLC*. Я применил указанную в статье команду

```
ffmpeg -i /path/to/source.avi -target pal-dvd /path/to/output.avi
```

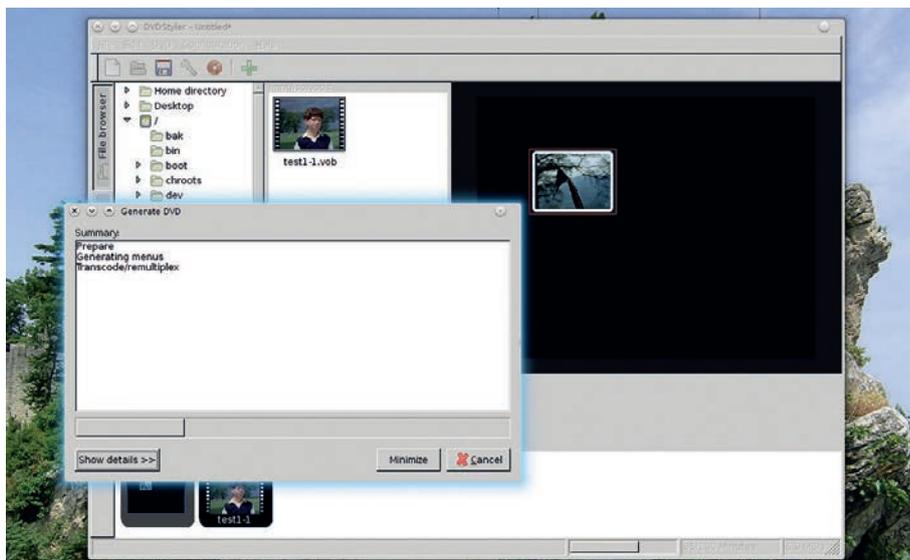
для получения «PAL-DVD совместимого файла». Что дальше делать с этим файлом? Я записал его на DVD с данными, и проигрыватель говорит, что на диске нет видеофайлов. Я попробовал создать видео-DVD в *Brasero*, поместив туда файл, но *Brasero* завис во время «преобразования файла в *mpeg2*».

Дэвид Хигнет [David Hignett]

Вы преобразовали свой ролик в MP4 и затем в AVI, но на DVD все видеофайлы должны быть в контейнерах MPEG2. Параметр `-target` в *Ffmpeg* гарантирует использование подходящего битрейта и кодеков, но тогда файл все равно помещается в контейнер AVI. Ничего этого делать не требуется, поскольку Ваш файл на исходном DVD уже именно в том формате, который нужен для DVD. Надо лишь извлечь этот файл. Существуют разные программы, которые это делают: например, *Vobcopy* (<http://vobcopy.org>) умеет извлекать отдельные треки, зашифрованные или нет, командой:

```
vobcopy -n X
```

где X — номер трека, который вы хотите скопировать. Порядок треков на диске и в меню может быть разным, но с помощью команды `lsdvd` можно просмотреть список треков с номерами и длиной, что должно помочь Вам определить нужный трек. Но сделать одно это не достаточно. Спецификация DVD требует определенной схемы расположения файлов на диске, даже если на нем



DVD, даже без меню, обязан иметь определенную структуру файлов. *DVDStyler* для вас ее создаст.

всего один фильм, который запускается автоматически, и нет меню. Сделать копию диска вместе со структурой меню поможет утилита *Dvdbackup* (<http://dvdbackup.sourceforge.net>), которая изначально и предназначалась для резервного копирования дисков; но с ее помощью можно также скопировать отдельные треки, сохранив правильную структуру файлов. Запустить ее можно, например, так:

```
dvdbackup -t X
```

Обе программы по умолчанию ищут диск в `/dev/dvd`; если он где-то в другом месте, воспользуйтесь параметром `-i`. Получив файл VOB, можно создать DVD с помощью *DVDStyler* (<http://www.dvdstyler.org>). Запустите программу и задайте в первом окне формат видео (PAL или NTSC). Если Вам требуется диск с одним роликом, который просто будет запускаться, пропустите раздел с шаблонами, нажмите кнопку + и добавьте свой файл. В окне открытия файла вам нужно будет выбрать просмотр всех файлов, так как по умолчанию VOB-файлы в нем не отображаются. Затем нажмите на иконку *Burn* [Прожечь], укажите путь для ISO-образа и подождите, пока он будет создан. Он создастся очень быстро, так как никакого перекодирования не производится. Получив файл ISO, его можно просмотреть в *VLC*, чтобы убедиться, что все в порядке, и затем записать на диск через *Brasero*.

2 Меню отмалчивается

В Мне не удалось установить никакой Linux на компьютер моей сестры (с уходом XP в отставку в апреле 2014 года, кажется,

пришла пора переменить ее с темной стороны). Компьютер у нее старый (примерно 2006 года) с процессором Celeron 3,06 ГГц и жестким диском в 80 Гб. Сначала я попробовал несколько live-дистрибутивов с дисков LXF, но не продвинулся дальше начального окна выбора дистрибутива. Я не могу переместить выделение в меню и не могу выбрать выделенный вариант нажатием Enter. Потом я загрузил openSUSE 13.1, но лучше не стало, разве что по умолчанию там выбран вариант загрузки с жесткого диска, с небольшим таймером. Таймер хотя бы движется — хотя и очень медленно — поэтому компьютер начинает загрузиться с жесткого диска, и мне не приходится выдергивать его из розетки.

Ну вот как перейти этот экран загрузки? Или современные дистрибутивы слишком сложны для архитектуры компьютера, на который я пытаюсь их установить?

суу13 с форумов

Дистрибутивы — и Linux вообще — здесь ни при чем: все дело в неработающей клавиатуре. Это особенно заметно на дисках *Linux Format*, так как на них нет дистрибутивов по умолчанию, которые загружаются, если Вы не выбрали ничего. Вам нужно зайти в настройки BIOS и включить USB-клавиатуру или устаревший режим USB [Legacy USB mode] (в разных BIOS терминология различается). После этого BIOS сможет распознать клавиатуру (я предполагаю, что у Вас USB-клавиатура). Это никак не повлияет на использование клавиатуры в системе:



» Когда вы видите окно загрузчика, ни одной операционной системы еще не загружено, и все устройства контролируются напрямую BIOS, а не ядром Linux.

мы просто сделаем ее доступной на ранних этапах загрузки.

Возможно, Вы заметили потенциальный недостаток этого совета: как же войти в меню настроек BIOS, если клавиатура не работает? Ответ — воспользоваться для этого клавиатурой PS/2, раздобыв либо старую клавиатуру, либо один из переходников с USB на PS/2, которые раньше поставлялись в комплекте с USB-клавиатурами. Опять же, это потребует только в единственном случае — для того, чтобы включить клавиатуру на время загрузки; впоследствии Вы сможете вернуться к своим привычным настройкам.

После того, как дистрибутив начнет загружаться, управление перейдет к драйверам USB ядра Linux, и проблема исчезнет. Именно это и происходило в Windows, в которой Вам не приходилось ничего выбирать при загрузке для запуска Windows.

3 Ложная интерпретация

У меня Raspberry Pi с ОС Debian Wheezy. Я написал программу на Python (`prog_control2.py`), которая прекрасно запускается, и хочу, чтобы она запускалась во время загрузки. Раньше у меня это получалось, но теперь нет. Скрипт я установил в `/etc/init.d/progstarter`.

Команда

```
sudo /etc/init.d/progstarter start
```

выдает сообщение «нет такого файла или каталога [no such file or folder]». Согласно `ls -l`, файл существует, он исполняемый, и его владельцем является `root`. Программа на Python находится в моем домашнем каталоге и тоже является исполняемой. Я проверил файл `progstarter`, и, насколько я вижу, ошибок в нем нет. После дополнительных проверок я обнаружил, что проблема связана со строкой `#!`. В режиме отладки появляется следующее сообщение:

```
-BASH: /etc/init.d/progstarter: /bin/sh^M
bad interpreter: No such file or directory.
```

И я озадачен, потому что `^M` — это, видимо, перевод каретки в конце строки.

Дон Стотт [Don Stott]

Вы правы, `^M` — это действительно символ возврата каретки (CR). Похоже, этот конкретный файл был создан на компьютере с Windows, так как в Windows для обозначения конца строки используется CRLF (LF — это перевод строки [Line Feed]). В Linux используется только LF, поэтому CR воспринимается как часть строки, т.е. последний символ, который должен обработать интерпретатор. Проверить это можно командой `file`:

```
file /etc/init.d/progstarter
```

Вы, скорее всего, увидите один из следующих вариантов:

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`.

POSIX shell script, ASCII text executable, with CRLF line terminators

POSIX shell script, ASCII text executable

Вам-то нужен второй вариант, но Вы увидите первый. Совет на будущее — создавайте файлы »



Коротко про...

X-проброс

X — основа всех графических интерфейсов Linux — это сетевой протокол. Хотя ваш монитор обычно связан с X-сервером на локальном компьютере, это вовсе не обязательно. Возможно, вы подключались к удаленным компьютерам через VNC или Nx, но их недостаток в том, что они полностью воспроизводят рабочий стол другого компьютера. Опять же, для этого на удаленном компьютере должен быть рабочий стол. Если удаленный компьютер — это автономный сервер, то рабочий стол — напрасная трата системных ресурсов. Вместо того, чтобы запускать рабочий стол целиком, можно запустить только ту программу, которая вам нужна, через SSH (это означает, что подключение будет безопасным даже в публичных сетях). В своем любимом терминале выполните команду:

```
ssh -X user@remote.computer someprogram
```

Если `someprogram` — графическое приложение, то на вашем рабочем столе должно открыться его окно. Здесь есть несколько моментов, о которых следует помнить. Во-первых, у вас должен быть доступ к компьютеру по SSH с паролем или ключом. На сервере SSH на другом компьютере должно быть включено перенаправление X (оно же — X-проброс). Обычно оно по умолчанию включено, но на всякий случай проверьте, что в файле `/etc/ssh/sshd_config` есть строка

```
X11Forwarding yes
```

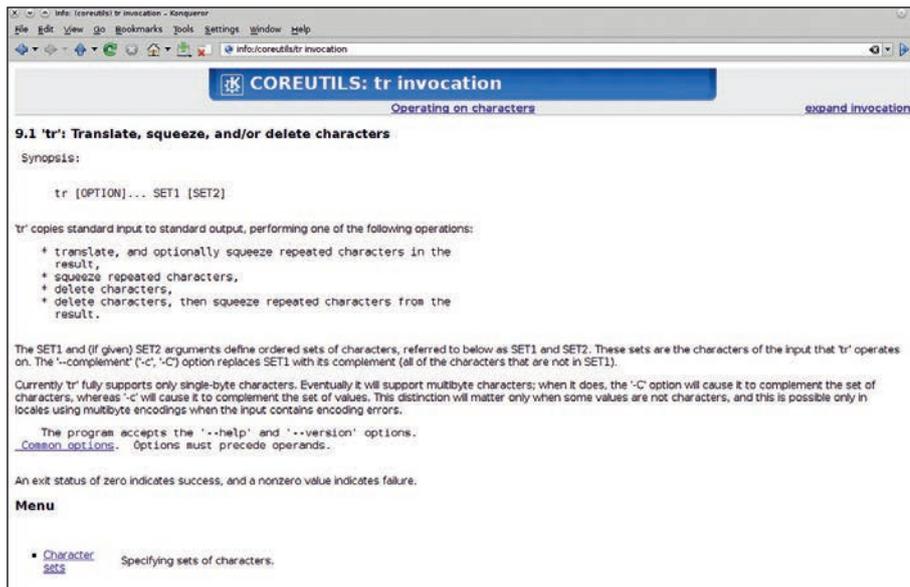
У `ssh` есть и другие параметры, способные вам пригодиться. Вместо `-X` можно использовать `-Y`. Он работает только с доверенным перенаправлением X11 [Trusted X11 Forwarding], но может быть заметно быстрее. Еще один параметр, способный заметно увеличить скорость при низкоскоростном подключении, это



» Запуск программы настройки сети Fedora на рабочем столе другого компьютера.

`-C`: он включает сжатие всех данных. Впрочем, для по-настоящему медленных подключений X не подходит, лучше будет взять `TightVNC` или `Nx`.

В MacOS для отображения видеоданных также используется X, поэтому вы сможете отображать окно с программой Mac на компьютере с Linux, и наоборот.



➤ Для преобразования файла Windows можно воспользоваться *Dos2unix*, *Recode* и даже старой доброй *tr*.

в редакторах Linux; ну, а сейчас Вам придется как-то конвертировать этот файл. Есть несколько вариантов, самый простой из которых — программа *Dos2unix*. Если она установлена, то преобразовать окончания строк к стандарту Unix можно командой

```
dos2unix /etc/init.d/progstarter
```

Другая программа для решения этой задачи — *Recode*, запустите ее так:

```
recode ibmpc.la1 /etc/init.d/progstarter
```

Какую из этих двух программ брать, зависит от того, какую Вы скорее найдете, а не от обширных технических различий между ними. Синтаксис проще у *Unix2dos*, но у *Recode* больше параметров. Альтернативный вариант — воспользоваться почтенной программой *tr*, которая входит в пакет *coreutils* и поэтому есть во всех дистрибутивах GNU/Linux. Она отправляет преобразованный текст из стандартного ввода в стандартный вывод, поэтому нужно применить перенаправление:

```
tr -d "\r" </etc/init.d/progstarter >/etc/init.d/progstarter.new
```

Кстати, создавать отдельный файл для запуска одной программы не нужно. В каталоге */etc/rc.local* есть скрипт, который запускается в конце процесса загрузки, и можно добавить свои команды в него. Не забудьте добавить *&* в конец команды, чтобы она освободила управление и не блокировала дальнейшей работы скрипта.

4 Для твердотельных

В Я только что купил твердотельный жесткий диск — как лучше всего настроить его в Linux? Какие настройки нужно использовать для разделов и установки границ разделов? [Ram, с форумов](#)

Особой разницы в настройке обычных и твердотельных дисков сейчас нет. Самым большим различием оставалось выравнивание разделов. Оно и по-прежнему остается, но в последних версиях *fdisk* и *parted*

по умолчанию используется безопасное выравнивание, которое работает с твердотельными дисками, а также с новыми жесткими дисками с размерами блока 4K. Ext4 — хорошая файловая система общего назначения, которая работает как с твердотельными, так и с обычными жесткими дисками, хотя в BTRFS есть несколько возможностей, специфичных для твердотельных жестких дисков. Хотите ли Вы доверить свои данные экспериментальной файловой системе, решать Вам. Последний раз, когда я беседовал об этом с кем-то из SUSE (а именно они являются главными поборниками BTRFS), они советовали использовать BTRFS для операционной системы, а ext4 — для данных, при этом речь шла о клиентах промышленного уровня, для которых целостность данных имеет первостепенное значение.

Некоторые продолжают считать, что большое количество операций записи с SSD может быть проблемой, но большинство последних исследований показывают, что твердотельный диск способен пережить обычный. Однако не будет лишним добавить опцию *noatime* к параметрам монтирования в */etc/fstab*. В большинстве дистрибутивов некоторые каталоги, куда часто производится запись, например, */tmp*, монтируются с файловой системой *tmpfs*. Поправка может производиться автоматически или вручную (обычно с помощью *Cron*). Чтобы она производилась автоматически, добавьте параметры *relatime, discard* к параметрам монтирования в */etc/fstab*, но не делайте этого для корневой файловой системы или любой другой файловой системы, в которую часто производится запись: сообщалось, что со временем это приведет к снижению производительности. Зато это прекрасно подходит для менее критичных и, главное, менее часто записываемых файловых систем. Более безопасной альтернативой будет запуск команды *fstrim* из *Cron*. Для приведения в порядок корневой файловой системы выполните команду

```
fstrim /
```

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 необходимо 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
```

Запускайте ее один-два раза в день, не чаще. Если Ваш компьютер круглосуточно включен, запланируйте ее запуск на то время, когда Вы им не пользуетесь. Вам понадобится отдельное задание *fstrim* в *Cron* для каждой файловой системы на SSD, которая не смонтирована с опцией *discard*. Если ни в одной из файловых систем параметр *discard* не используется, можно просто ограничиться командой

```
fstrim --all
```

Она молчаливо проигнорирует все файловые системы, которые не поддерживают *discard*, поэтому неважно, если при запуске команды у вас будет подключен внешний USB-диск. Если добавить параметр *-v*, *fstrim* сообщит, что она сделала. Это сообщение можно отправить на электронную почту, если в Вашем *crontab* установлен параметр *MAILTO*.

5 Исчезающие PDF

В В мою подписку на LXF входит загрузка из архива журналов на сайте. Однако при использовании *Firefox 24.3.0 esr* и Linux LXLE ссылки для загрузки не работают, хотя войти в систему удается. Я обнаружил, что *Firefox* тайно загружает PDF-файлы и помещает их в */tmp*. У этих файлов расширение *.part*, которое сохраняется даже после окончания загрузки



➤ С неверно заданным типом MIME Firefox может попытаться открыть PDF-файлы с сайта нашей подписки, вместо того, чтобы загружать их.

файлов. Я настроил Firefox так, чтобы он спрашивал меня, куда сохранять файлы, и для других сайтов это работает.

Я заметил разницу между вашим сайтом и другими сайтами, которые я пробовал. На них — прямые ссылки на PDF, а на вашем используется PHP. Не знаю, имеет ли это отношение к делу.

Хью Мак-Каллок [Hugh McCullough]

○ Подозреваю, что разное поведение при загрузке PDF-файлов на нашем сайте и на других имеет отношение к типу MIME, отправляемому в заголовке HTML, но у меня нет доступа к коду PHP, так что я не могу это подтвердить. Прямые ссылки не используются

потому, что файлы должны быть доступны лишь подписчикам, а значит, нужен код, который проверяет, зашли ли Вы на сайт (я знаю, что эту проверку можно выполнить и по-другому, но таким образом она реализована на данный момент).

Странные имена файлов в /tmp могут означать, что Firefox пытается скачать файл и создает временный файл в /tmp, который Вы видите. Однако у меня не получилось воспроизвести Вашу проблему, даже в том же дистрибутиве и с той же версией Firefox. Это может означать, что Ваша проблема вызвана настройками или расширениями Firefox.

Вы можете отключить все расширения и темы, запустив Firefox из терминала в безопасном режиме:

```
firefox --safemode http://www.linuxformat.com/subsarea
```

Если это не поможет, попробуйте запустить Firefox с совершенно чистым профилем. Выйдите из Firefox, переименуйте .mozilla в .mozilla.bak и снова запустите Firefox. Будет создан новый каталог .mozilla, но когда Вы закончите, его можно будет спокойно удалить и заменить на .mozilla.bak. Если проблема не проявилась, когда выполнялся запуск без расширений, включайте их одно за другим, пока не обнаружите виновника. Аналогично поступите, если проблему помог решить чистый профиль: Вам придется поэкспериментировать с настройками, чтобы найти неправильно заданный параметр. Также может поспособствовать запуск Firefox из терминала, а не с рабочего стола, поскольку тогда Вы увидите все сообщения, видимые Firefox.

6 Не работает Ubuntu?

В Запустив Ubuntu 13.10 Live с диска LXF183, я вижу графический рабочий стол, но клавиатура и мышь не работают. Мой компьютер зависает, и приходится выдергивать его из розетки — не помогает даже сброс компьютера кнопкой. Проблема в моем DVD или есть мне не хватает какой-то функции? Другие дистрибутивы — Mageia 4 и openSUSE 13.1 — прекрасно работают на моем настольном компьютере Dell E521.

Питер Лонсдейл [Peter Lonsdale]

○ Проблема вряд ли в диске, но это легко проверить. Смонтируйте DVD, откройте терминал и выполните команду

```
sh /path/to/dvd/media-check.sh
```

При любых ошибках на диске команда сообщит о них. Скорее всего, компьютер не зависает — просто перестают работать клавиатура и мышь. Чтобы это проверить, подождите минуту и посмотрите, изменится ли время на часах. Было несколько сообщений о том, что мыши и клавиатуры с USB-интерфейсом не работают с последней версией Ubuntu (на LXF183 есть обновленная версия Ubuntu). Это относится к нескольким USB-чипсетам и устройствам и вызвано отсутствием драйверов, а поэтому не влияет ни на одну из систем, которые мы проверяли на данном DVD. Попробуйте другую клавиатуру и мышь или подключите ту же клавиатуру к другому USB-порту. Если это не поможет, то единственная альтернатива — использовать мышь и клавиатуру с интерфейсом PS/2. После установки Ubuntu обновление системы должно помочь горю. LXF



Часто задаваемые вопросы

Wine

➤ Что такое Wine?

Это аббревиатура фразы "Wine Is Not an Emulator [Wine — это не эмулятор]".

➤ А если не эмулятор, то что же это?

Ну, вообще-то — своего рода эмулятор. Wine — открытая реализация API Windows поверх X и Unix.

➤ С Wine можно запустить Windows в Linux?

Не совсем. Wine — слой совместимости между программами Windows и Linux. Когда вы запускаете программу Windows в Wine, программа думает, что она работает в Windows, но вызовы функций Windows, которые она выполняет, преобразуются в вызовы функций Linux.

➤ Нужна ли установленная Windows, чтобы пользоваться Wine?

Нет, но если Windows установлена, Wine воспользуется ее DLL-библиотеками. Если нет, у нее есть альтернативные варианты.

➤ Если у меня уже есть Windows, зачем мне Wine?

Прежде всего затем, чтобы можно было запустить программу Windows, не перезагружаясь в Windows и обратно в Linux. То есть, программы Windows можно запускать в Linux вместе с программами Linux. С Wine также можно пользоваться плагинами Windows для программ Linux, таких как web-браузеры и медиа-проигрыватели. Это дает программам Linux доступ к форматам, поддерживаемым только проприетарными кодеками Windows.

➤ Бывает, что эмуляторы тормозят — а насколько быстр Wine?

Wine эмулирует только API Windows, а не устройства. А значит, программы в Wine будут работать с той же скоростью, что и в Windows, иногда даже быстрее. Недостаток — Wine может запускаться только на том «железе», что и Windows; это не эмулятор процессора.

➤ Как ее загрузить?

Загрузить Wine во множестве форматов можно с сайта Wine. Пакеты также доступны в репозиториях большинства дистрибутивов. Еще есть две коммерческих версии Wine. CrossOver Office от Codeweavers предназначена для запуска рабочих приложений Windows и может запускать плагины Internet Explorer в некоторых браузерах для Linux. Вся сложная настройка выполнена за вас. На сайте Codeweavers перечислено 50 поддерживаемых приложений, но на самом деле их гораздо больше.

➤ С рабочими приложениями понятно, а как насчет игр?

Другая коммерческая версия — Cedega, ранее известная как WineX. Cedega от TransGaming Technologies — улучшенная версия Wine с поддержкой DirectX. Cedega доступна бесплатно через репозиторий CVS. Если вам нужны готовые пакеты и поддержка, надо будет оформить подписку. Список поддерживаемых программ весьма обширен, и он постоянно пополняется.

➤ А откуда ее загрузить?

Wine есть в репозиториях большинства дистрибутивов. Исходные коды и пакеты для большинства дистрибутивов можно найти на www.winehq.com. CrossOver Office, включая бесплатную пробную версию, доступен на www.codeweavers.com, а Cedega — на www.transgaming.com.



LXF HotPicks



Ричард Смедли

Активист FOSS, Ричард колесит по вязким полям Interweb на своем верном SMED5000 в поисках новых залежей сокровищ открытого кода для вас.

Midnight Commander » Python 3.4 » Goggles Music Manager » Nuttx » Qtractor
» Shellpic » Dispcalgui » Pioneer » Assassin of the Underdark » Glucat » Git

Менеджер файлов командной строки

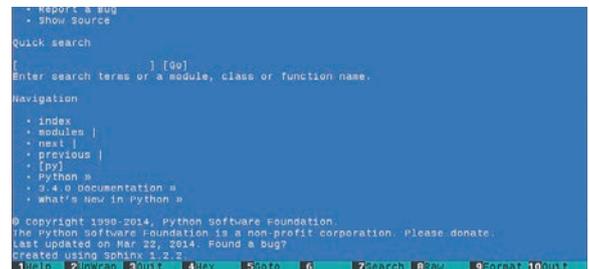
Midnight Commander

Версия: 4.8.12 Сайт: www.midnight-commander.org

Как ни странно, есть люди, не испытывавшие чистую радость от использования *Midnight Commander*. Это полезное и не требующее лишних усилий приложение — вызываемое из терминала другим своим именем, *mc* — работает там, куда не ступала нога менеджеров GUI, скажем, в сессии терминала SSH, и дает вам удобство в работе с файлами и папками в классическом двухпанельном интерфейсе с помощью отточенного временем набора горячих клавиш. Мигель де Икаса [Miguel de Icaza] разработал *mc* в 1994 году, еще до Gnome и Mono, и его долготелie — залог того, что в репозитории вашего дистрибутива, вероятно, уже есть

свежий пакет; но компиляция *mc* тоже отнюдь не проблема. Просмотрите исходный код: это очень читаемый, зрелый проект. И в каждом релизе имеются новинки — здесь это дружелюбная программа выбора скинов и улучшенная поддержка *zsh*.

Через свою основную программу, *mcedit*, и вспомогательные приложения *mc* может читать, редактировать и манипулировать файлами в локальных и удаленных



» *Midnight Commander* вызывает вспомогательные приложения, например, браузер командной строки *Lynx*, чтобы просматривать файлы всех типов.

«*mcedit* — достойнейший редактор, применимый и отдельно.»

папках, даже если те упакованы в tar-архиве. Некоторые дистрибутивы в качестве редактора по умолчанию в *mc* используют *nano* — его можно сменить на *vi* или *Emacs*; но *mcedit* — достойнейший редактор, применимый и отдельно, с подсветкой синтаксиса для всего, от Cython до Verilog.

Исследуем Midnight Commander

Внутренние архивы

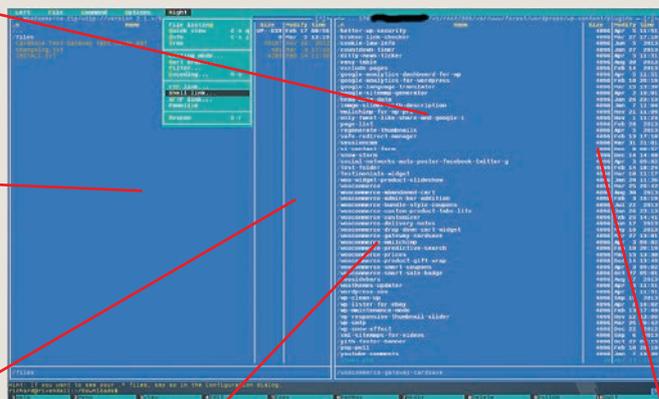
Работать в архивированной папке легко: *mc* распознает большинство форматов архивирования.

Меню и опции

Выпадающие меню позволяют проводить дополнительные операции с файлами и выдают доступ к опциям настройки.

Локально или удаленно

Используя SSH, FTP или SFTP, *mc* работает с локальными и удаленными файлами, позволяя передать файл одним нажатием клавиши.



Ярлыки функциональных клавиш

Каждая функциональная клавиша — это ярлык команды или меню; некоторые изменяются контекстуально, так что подсказка не помешает.

Опция Edit

Edit — постоянно присутствующая опция: как-никак, это менеджер файлов *nix. Редактор, естественно, можно прицепить любой, но *mcedit* — хороший выбор по умолчанию.

Чудо Web

Сессия *mc* зачастую быстрее, чем у альтернатив — например, рекурсивного *wget* на сайте — и теперь можно снова по всему сайту с помощью *mc*, просматривая код в *mcedit* или читая страницы в *Lynx* (или любом другом браузере, который вы включите в свой *mc*).

Как видно на скриншоте слева, в нижней строке — 10 пронумерованных комбинаций функциональных клавиш; напоминание нужно, поскольку часть из них зависит от контекста. Так, F2 обычно вызывает пользовательское меню, но в *mcedit* это клавиша Save. Если ваш менеджер окон перераспределит клавиши, можно жать Esc и номер клавиши; Esc 0 соответствует F10 (проверено в виртуальной консоли, и терминале ssh, и менеджере окон *XMonad*, где доступны все клавиши F). Но некоторые программы удаленного доступа не передают клавиш F: в клиентах SSH на мобильных проще использовать клавиши Esc.

Высокоуровневый язык программирования

Python

Версия: 3.4 Сайт: www.python.org

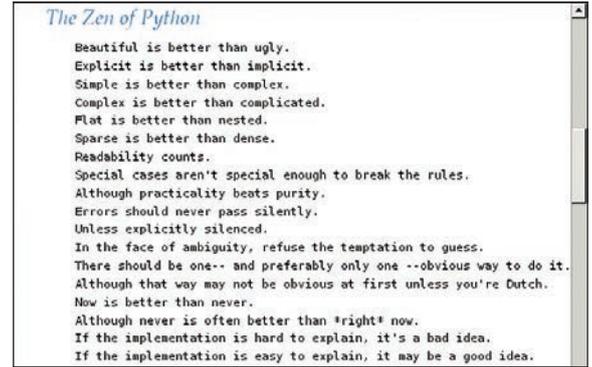
Спустя более полугодия после первого alpha 3.4 вышел полный релиз. Ничего сногшибательного: несколько новых функций, мелкие изменения. Давайте сперва коснемся именно их. Основное изменение — стандартизированный модуль *enum* для стандартной реализации перечисляемых типов — то есть отныне можно связать символические имена (например, дни недели) с константами. Такое предлагали еще в 2005 году, да и на сей раз не обошлось без возражений — число сообщений в списке рассылки разработчика перевалило за 1000. *Enum* обозвали неловким компромиссом, но большая часть сообщества поддержала его включение, и, похоже, решительно настроена одолеть любые проблемы, способные здесь возникнуть.

Не столь спорный асинхронный ввод/вывод наконец-то вошел в основную библиотеку, вместе с идеями, почерпнутыми из модуля *Twisted* и прочих источников.

Это исключительно важно для разработчиков облаков, и языковые умники явно расценят реализацию как потрясающую.

Любимчик учителя

Python широко применяется в образовании, и главные разработчики Python получили массу откликов от профессионалов в области образования по поводу проблемных зон языка для кодеров-новичков; общей жалобой стало отсутствие менеджера пакетов по умолчанию, это создает проблемы для кросс-платформенных приложений, которые менеджеры пакетов дистрибутивов Linux не могут преодолеть. Начиная с версии 3.4, программой по умолчанию для установки пакетов Python будет сделан *pip*, что и решит вопрос.



Python развивается, но введите `import this` в строке приглашения, и вы по-прежнему увидите юмористические напоминания о его принципах.

После многих отвергнутых предложений, за официальную высокоуровневую, объектно-ориентированную библиотеку для оценки локальной файловой системы приняли *pathlib*. Имеется также новый модуль статистики, в качестве легковесной альтернативы *NumPy*.

Но вот что действительно важно в этом релизе. Python 3 постепенно завоевывал пользователей — и завоевал их столько, что лишь несколько проектов остались чистым Python 2 (хотя, естественно, никуда не делась солидная кодовая база для проектов Python 2). Учитывая скорость перехода на Python 3, мы рассудили, что за период действия Python 3.4 мы увидим, как Python 3 наконец станет уверенным выбором по умолчанию.

«Увидим, как Python 3 станет уверенным выбором по умолчанию.»

Менеджер музыкальной коллекции и плеер

Goggles MM

Версия: 0.13.1 Сайт: <http://gogglesmm.github.io>

Goggles Music Manager — быстрый легковесный музыкальный менеджер, обновленный год спустя. Среди изменений — новый движок воспроизведения и улучшенный интерфейс пользователя, менеджер подкастов и большая точность поиска по FLAC, Vorbis, Opus и MP4. Однако изменения в движке оставили AAC в стадии незавершенки, что исключит его из рассмотрения тех, кто использует эту схему сжатия и кодировки.

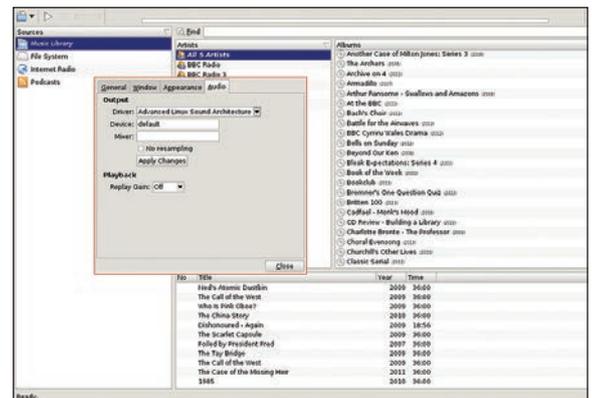
Вам нужно будет скачать набор инструментов *FOX*, если только у вас нет его последней версии для какого-то другого проекта. После этого запуск `./configure --without-oss` и установка недостающих пакетов разработчика (*libglew* и *libcrypt*) обеспечили нам компиляцию без ошибок, а после `make` и `make install` мы открыли *GogglesMM* из меню приложений и были спрошены о выборе директорий для импорта — по умолчанию это `~/Music`. Когда мы это приняли, *Gogglesmm* быстро

прочесал директорию рекурсивно и добавил в свою библиотеку нашу коллекцию подкастов и скачанных с *iPlayer* файлов, вместе с какой-то странной музыкой.

Менеджер-минималист

Категоризация автоматическая, по жанру, исполнителю, альбому и песне. Это не всегда удобно, если ваша музыкальная коллекция выходит за рамки привычного и популярного или если импортируемые папки не имеют подходящих названий, но здесь поможет простое редактирование тэгов *GogglesMM* — которое поддерживает работу с несколькими файлами.

Интерфейс пользователя, созданный с помощью набора инструментов *FOX*,



Хороший менеджер для вашей музыки — а также постановок, аудиокниг, подкастов с вашего жесткого диска.

радует минимализмом: все, что нужно, найдется или в трех выпадающих меню, или по щелчку правой кнопкой на разных пунктах. Где можно, отображается обложка альбома (или подкаста). Подкасты и интернет-радиостанции можно добавлять по URI, а плей-листы — импортировать или экспортировать в форматах XSPF, M3U, PLS или CSV. Плей-листы можно реорганизовать перетаскиванием или рассортировать по опциям меню. *GogglesMM* также поддерживает *last.fm* и *libre.fm*, со ссылками на страницы регистрации, если у вас нет учетной записи. Нажав на `Ctrl+M`, вы получите версию «миниплеера» пользовательского интерфейса.

«Прочесал директорию рекурсивно и добавил нашу коллекцию.»

Встроенная операционная система

NuttX

Версия: 7.1 Сайт: www.nuttX.org

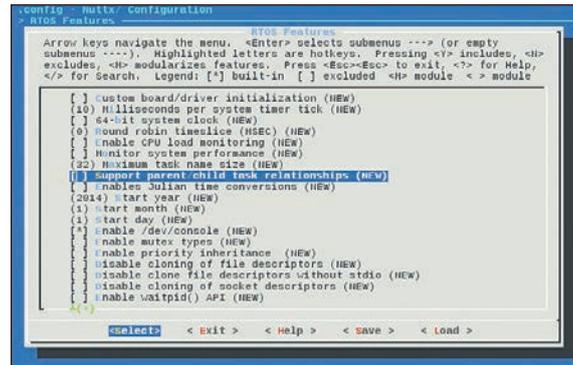
Специализируясь на самых маленьких, глубоко встроенных средах, NuttX — это интересная встроенная операционная система в реальном времени (RTOS) для работы на встроенных платах разработчика, которые накопились у вас дома или на вашем хакерском рабочем месте. Фактически, NuttX скорее всего сработается с большинством имеющихся у вас плат, поскольку это — NetBSD встраиваемого мира, охватывающая все, от Atmel 8-bit AVR до Zilog Z80, не брезгая ARM или Hitachi SuperH.

При наличии соответствующей платы вы можете использовать *qemu-system-arm* (или как там называется ваш процессор), чтобы протестировать программы NuttX, прежде чем напрягать ПЗУ на плате. Однако для тестирования и разработки NuttX нужно установить инструментальный кросс-компилятор *toolchain*. Кроме NuttX 7.1, придется скачать еще и отдельный пакет, с именем *apps-7.1*, который содержит

необходимые вам программы — оболочку, сетевые программы и все прочее.

Ностальгия по ядру

NuttX использует *kconfig-frontent*, для которого вам понадобится *flex* и обычные основные программы, а также *gperf*. О большинстве из них позаботится ваш менеджер пакетов, но *kconfig-frontent* придется скомпилировать свежий; мы взяли 3.12.0. Запустив *make menuconfig*, вы увидите экран, который вернет вас во времена, когда вы сами компилировали свое ядро Linux — особенно если у вас была система со слегка нестандартным оборудованием. А если честно, в те дни почти все оборудование считалось нестандартным, особенно ноутбуки.



➤ Когда ж это вы в последний раз компилировали ядро? NuttX освежит ваши воспоминания благодаря *kconfig-frontent*.

Сейчас мир ПК стал добрее, но с таким количеством опций встраиваемых чипов вам придется-таки потрудиться — и, к сожалению, ввиду изобилия этих опций, мы не можем предложить вам руководство. Вместо этого мы укажем вам на залитые солнцем предгорья успешной разработки и посмотрим, что получится, когда вы поставите NuttX на плату; например, на нескольких платах с 16 КБ ОЗУ установка NuttX и *NSH* оставила свободными более 10 КБ. К тому же среда очень похожа на Linux, так что вы быстро привыкнете.

«Вернет во времена, когда вы сами компилировали ядро Linux.»

Многодорожечный секвенсор Audio/MIDI

Qtractor

Версия: 0.6.0 Сайт: <http://qtractor.sf.net>

Многодорожечный audio/MIDI секвенсор, написанный на C++ в среде Qt4, *Qtractor* использует *Jack* для аудио и *ALSA* для MIDI, и его цель — стать полноценной домашней студией. На данном этапе пакеты имеются для Arch, Fedora и SUSE.

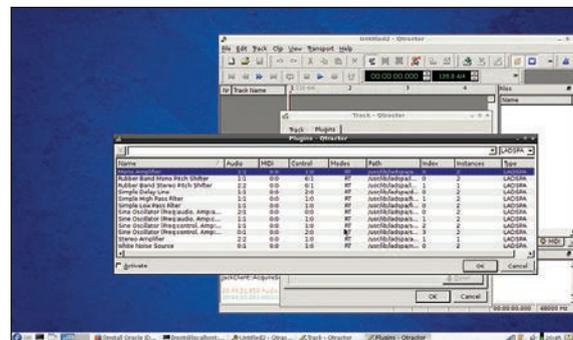
Если вы вынуждены компилировать *Qtractor* сами, обязательных зависимостей немного — *Jack*, *ALSA*, *libsndfile*, *LADSPA* и *Qt4*. Но, ограничившись этим буквенным вишнегретом, вы пропустите ряд отличных технологий: например, *LV2*, расширяемый преемник *LADSPA*, который *Qtractor* начал использовать одним из первых, плюс *librubberband*, *Rubber Band Audio Time Stretcher* и API *DSSI* для плагинов синтезатора. Загляните в список дополнительных требований к компиляции, имеющийся на сайте проекта, и установите побольше.

Под скромным номером версии 0.6.0 *Qtractor* прячет стабильность и зрелость. Время ожидания невелико, и — частично

благодаря *Jack* — связь между реальными и виртуальными инструментами и микрофонами в программе отличается гладкостью. Но это при компиляции с *DSSI*.

Отношение не включено

Простейший способ запустить его — щелкнуть правой кнопкой по области файлов (панель в правой части окна), чтобы выбрать музыкальные файлы, с которыми вы будете работать. *Qtractor* — безопасный секвенсор для композиции и аранжировщик, так что он не будет изменять ваши файлы. Для большей части задач вам, вероятно, понадобятся плагины помимо уже включенных. К счастью, поддержка плагинов *LADSPA*, *DSSI*, *Linux VST* и *LV2* здесь



➤ *Qtractor* обладает полезным, хоть и небольшим, набором плагинов по умолчанию, поддерживаемых сторонними плагинами.

хорошая. Можно также использовать стандарт плагинов Steinberg VST с *Wine*, употребив надстройку *DSSI-VST*. 110-страничное руководство было обновлено по версии 0.5.x, но это неплохая стартовая площадка, и в него включен следующий перл: «Отношение. Необязательный компонент, но если вы намерены стать рок-звездой, вам стоит поработать над бунтарским отношением к миру и решительно отвергнуть программы из мейнстрима».

Qtractor отличается интересным противоречием: он рассчитан на любителей, крошечную область влияния и ориентированность исключительно на платформы Linux, но мог бы преспокойно конкурировать даже с некоторыми профессиональными аудиопакетами.

«Мог бы конкурировать с профессиональными аудиопакетами.»

Просмотрщик изображений в терминале

Shellpic

Версия: 1.3 Сайт: <https://github.com/larsjsol/shellpic>

Shellpic использует коды переключения оболочки для показа изображений в терминале или клиенте IRC. Да, именно, не произведения ASCII-арта, а настоящие картинки; правда, несколько угловатые, но зато напоминающие старые добрые 8-битные денки (если, конечно, у вас дома не стоял монохромный компьютер).

Возможность отправлять картинки в IRC — более чем новшество, хотя его удобство ограничено низким разрешением. Тем не менее мы убили немало времени, разглядывая фотоальбомы и старые фотографии в их славном угловатом преломлении. Одни были не вполне узнаваемы, а другие выходили неплохо — скажем, четкие снимки архитектурных сооружений или людей в ярком глубоком цвете.

Недавние промежуточные релизы Shellpic добавили поддержку анимированных изображений, а последняя версия произвела дальнейшие усовершенствования. 1.3 также добавляет поддержку

16-цветных терминалов; хотя по умолчанию цветов 256. Если ваш терминал имеет 24-битную цветовую поддержку, параметр `--shell24` включит 24-битный результат.

При его запуске как `shellpic <pic.file>` Shellpic пытается вписаться в ваше окно терминала. В виде альтернативы можно указать, масштабировать ли по размеру или не надо. Прочие параметры включают зацикленную анимацию, глубину цвета и создание изображения, подходящего для передачи в клиент IRC.

Простая установка

Хотите попробовать? У вас должен быть установлен *pillow*, который ваш дистрибутив может включить в пакеты как *python-pil*. Теперь — или

«Старые фотографии в славном угловатом преломлении.»



» Париж весной — однако рендеринг его в окне терминала лишний раз подтвердит экзотичность любителей командной строки.

```
sudo pip install Shellpic
если вы работаете в стиле Python, или
git clone https://github.com/larsjsol/shellpic.git
cd shellpic
sudo python setup.py install
```

чтобы найти самую свежую версию.

Мы устанавливаем для раздела HotPicks множество программ — некоторые из них требуют ужасной мороки и на эти страницы не попадают; но надо сказать, что при виде проекта, написанного на Python, мы вздыхаем облегченно, поскольку обычно они устанавливаются и работают без особых проблем. Мы склонны избегать неблагоприятных сравнений с другими языками; но, кодеры Python, похлопайте друг друга по спине.

GUI для утилит калибровки отображения Argyll CMS

DispcaGUI

Версия: 1.9.0.6 Сайт: <http://dispcaGUI.hoech.net>

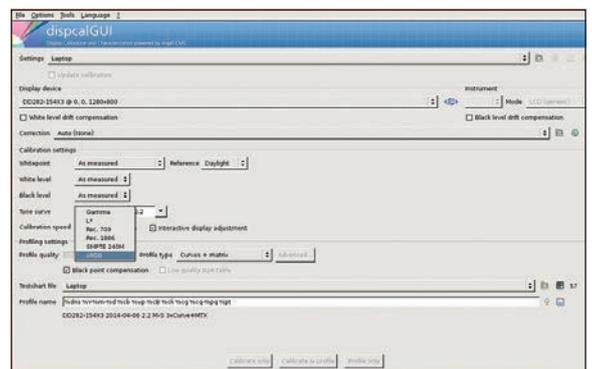
Читатели, которые проводят время вне командной строки и работают с фотографиями на экране, не могут не заметить угнетающее искажение цветопередачи при переходе с ЭЛТ-монитора на ЖК-дисплей. Но не все потеряно! Любой дисплей поддается калибровке. В Linux управление цветом использует профили Международного консорциума (ICC). Ваше устройство снабжается точным цветовым профилем, конвертируемым для поддержания правильной цветопередачи, от сканера или камеры через программы редактирования изображений до монитора или даже принтера. Применение ICC обходит проблему отсутствия драйверов для специальных колориметров и спектрофотометров в Linux: профили ICC годятся везде. Есть и другие опции, но самой верной ставкой на систему управления цветом (Colour Management System, CMS) в Linux будет Argyll CMS, от Грэма Гилла [Graeme Gill]. Этот набор инструментов включен

в пакеты у большинства дистрибутивов и неплохо обслуживается, но он чисто для командной строки, а при таком множестве сложных задач и выборов по калибровке GUI облегчил бы жизнь львиной доле обычных пользователей.

Много цветов

DispcaGUI и есть этот графический интерфейс для инструмента калибровки Dispca Argyll CMS и ряда его инструментов профилирования. Монитор должен быть первым в очереди по настройке работы с цветами: ведь откуда бы ни прибыли ваши изображения, видеть вы их будете на экране. Вам, как пользователю, надо будет выбрать настройки для яркости,

«ICC обходит проблему отсутствия драйверов для колориметров.»



» Одно то, что это GUI, еще не означает, что тут ничего подстраивать — но это не сложнее цветовой калибровки.

гаммы и белой точки, применив сперва настройки монитора, а затем кривые калибровки видеокарты. Это не обеспечит точное отображение цветов, но даст лучшие результаты в приложениях без управления цветом. Прежде чем ко всему этому приступить, нужно настроить свой колориметр, в том числе настройку разрешений, чтобы заставить программы пользовательского режима говорить с оборудованием. К счастью, есть документация, покрывающая большинство случаев, а сайты вроде <http://dpreview.com> размещают руководства и форумы-обсуждения по использованию Argyll CMS и DispcaGUI.

HotGames Развлекательные приложения

Космические приключения

Pioneer

Версия: 20140402 Сайт: <http://pioneerspacesim.net>

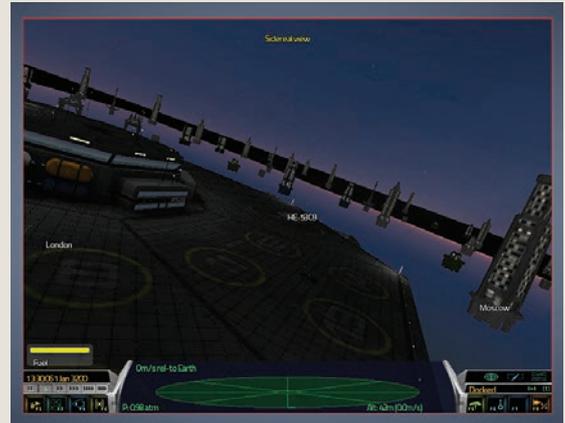
В духе *Frontier: Elite II*, *Pioneer* размещается в нашей галактике на рубеже XXXI века. Здесь много путешествий, что отражается в сложности интерфейса пользователя для полетов и разных видах карт. Вы сканируете информационные доски на торговых станциях, выбираете миссии; перевозите грузы; а периодически попадаете в беду, и вам приходится сражаться. Вы вольны заняться контрабандой или стать индустриальным преступником, но на ваши нарушения последует реакция закона.

Распакуйте бинарник и запустите его; всплывающий экран предложит вам подождать, пока он имитирует 13,7 миллиардов лет эволюции вселенной. После чего перед вами появится стартовое меню с возможностью выбора Земли; звезды Барнарда (привет, H2GT2G!);

или New Hope у Эпсилона Эридана, столицы Содружества Независимых Миров; а также опции настройки. Выберите стартовую точку — и вернетесь к руководству, поскольку это одна из тех всепоглощающих игр, где перед погружением предстоит многое изучить, и это не та игра, чтобы можно было «взять телефон и поиграть минут пяток».

Естественно, начни мы с чтения руководства, мы бы знали, что не стоит стартовать с дока тюрьмы строгого режима Tranquility на орбите звезды Барнарда, где нам дали «плохо оснащенный Xylophis, и это считается трудным стартом».

«Вы вольны стать преступником, но последует реакция закона.»



» Звезда Барнарда может казаться неплохим местом для старта — но тогда вам дадут плохо оснащенный корабль, и притом это тюрьма.

Начинать с Лондонского дока на Земле явно «считается легким стартом».

Читайте руководство — и следуйте инструкциям по полету из wiki — и ваш старт будет лучше нашего. Однако встретиться нам не суждено, потому что это не многопользовательская игра онлайн. Тут есть свои преимущества (помимо того, что вы не столкнетесь с эксцентричным стилем игры кое-каких онлайн-игроков) — вы сможете быстро промотать многие перемещения так, как вряд ли удалось бы в мире с общим доступом.

Исследование подземелий

Assassin of the Underdark

Версия: 0.3.15 Сайт: <http://bit.ly/SdCT5X>

Хотя по жизни мало кто отважится спуститься в темный и полный опасностей подземный мир ради призрачного шанса вознаграждения, это очень привлекательный формат для ролевых игр — как компьютерных, так и иных. *Assassin of the Underdark* — незаконченная игра, использующая библиотеку *rugame*, и сейчас ей повезло: свою деятельность возобновил Йохан Сойппенс [Johan Ceuppens], который занимается также проектами типа стрелялок с боковой прокруткой и клона *Super Mario Brothers*.

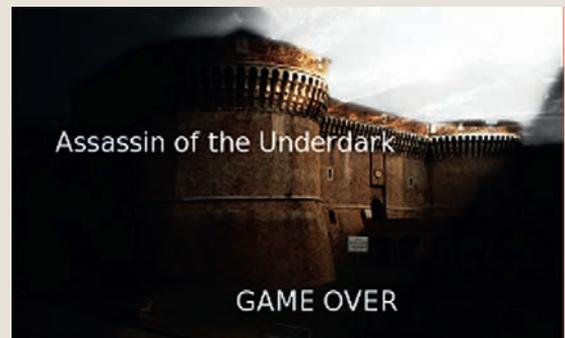
Стадия разработки на деле очень ранняя: «Реализовано две комнаты, вы начинаете игру с запасом бомб и можете сражаться мечом. Есть движок, и уже есть гоблин, наблюдатель, охранник и змеи».

Тем не менее, установка проста. Убедитесь, что у вас есть *rugame* и Python 3, затем скачайте, распакуйте и запустите игру командой `python game.py`

Вы увидите, что вам нужен персонаж (на данный момент это процесс в два щелчка), затем наступает время весьма рудиментарной схватки в подземелье, пока враги вас не осияют.

В чем суть, спросите вы? Ну, не исключено, что однажды эта игра станет столь же сложной и требовательной, как *Pioneer* (выше), но пока что в ней ничего

«Проект открытого кода на ранней стадии разработки.»



» Хотя игра и короткая, есть масса 00-кода для добавления и разработки, если вам охота поучаствовать в проекте Python.

особенного не происходит, и в этом вся прелесть. Не то чтобы ее доделали за пару минут, но это проект открытого кода на ранней стадии разработки. Изучаете ли вы программирование игр на Python (grep'ните код для FIX), сочиняете ли музыку или оттачиваете свои художественные навыки на компьютере, проект на ранней стадии развития дает вам отличную возможность присоединиться к нему ради всеобщей пользы. Вы проверяете свои силы в реальном совместном проекте, проект получает помощника, и в него вливаются новые идеи и умения.

Библиотека классов шаблонов

GluCat

Версия: 0.7.3 Сайт: <http://glucat.sf.net>

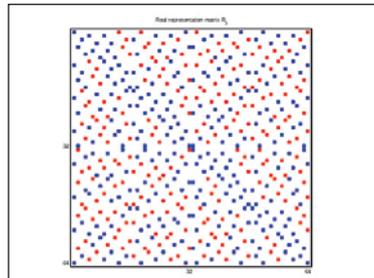
Generic Library of Universal Clifford Algebra Templates, или *GluCat* — это библиотека классов шаблонов, которая (оx!) «моделирует универсальные алгебры Клиффорда над полем вещественных чисел произвольной размерности и сигнатуры. *GluCat* реализует модель любой алгебры Клиффорда, соответствующей любой невырожденной квадратичной форме, до максимального числа измерений».

GluCat включает модуль расширения *PyClical*, который «реализует `index_set` классов Python и `clifford`, с интерфейсом к соответствующим классам C++ в *GluCat*».

Алгебра Клиффорда — которая требует большого объема матричных операций — имеет приложения в дифференциальной геометрии, квантовой теории и машинном зрении, где применяется к распознаванию действий. *GluCat* 0.7.2 (и исправление ошибок 0.7.3) — первый релиз после обновления C++11 пару лет назад, которое также включало набор

руководств, объясняющих работу с алгебрами Клиффорда в Python. Данный релиз генерирует блокноты IPython, ускоряет некоторые функции, и теперь компилирует и делает сборку с помощью `g++ 4.8` и `clang++ 3.4`.

Вам понадобятся библиотеки C++ Standard Library и Boost Library с той же версией `g++`, которую вы будете использовать для *GluCat*. Следуйте отличной документации, и вы мигом начнете добавлять быстрые шаблонные классы к своему проекту C++ или Python.



Хотя представления матриц иногда смотрятся ситуативно, математика, применяемая *GluCat*, имеет приложения в компьютерном распознавании образов.

Распределенная система контроля версий

Git

Версия: 2.0 Сайт: <http://git-scm.com>

Для кодеров — кроме Python 3.4, еще одна большая новость о релизе: *Git* достигла уровня 2.0. Эта распределенная система контроля версий и управления исходным кодом очень быстро, начиная с ухода Торвальдса на *Git* взамен *BitKeeper* как держателя исходника Linux в 2005 году, выросла до такого состояния, когда мы воспринимаем ее как нечто само собой разумеющееся — высшая форма комплимента для любого инструмента.

Этот релиз, хотя и обозначен знакомым номером, не вносит особо потрясающих изменений в API или революционных функций, но меняет поведение по умолчанию. Главное изменение — переход семантики соответствий к простой семантике для `git push [S]there`. Если не указано разветвление, все соответствующие ветки больше не отправляются на удаленное расположение; текущая ветка отправляется на ветку с тем же именем. Также `git add`

`<path>` теперь то же самое, что `git add -A <path>`, так что `git add dir/` отмечает и записывает удаленные пути, приводя в соответствие `add` и `commit`.

Если вы им уже пользуетесь, все, что вам нужно знать — есть ли у вас серьезные причины обновлять его именно сейчас, а не позже? Не вредно и подождать, пока ваш дистрибутив обеспечит вас пакетом обновления: вы лишитесь разве что незначительных улучшений в работе, функциях и производительности. **LXF**

Когда мы выжили в печать, официального релиза не было, но анонс релиза сильно обновился.



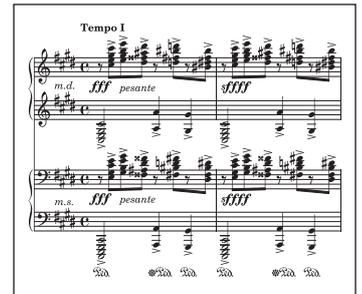
Также вышли

Новые и обновленные программы, тоже достойные внимания...

GNU Guile 2.0.10

Реализация Scheme с модульной системой, системными вызовами POSIX и множественными потоками.

www.gnu.org/software/guile



GNU Guile — Scheme в реальном мире, движущая LilyPond.

JERL r9.1 5.18.2

Запускайте скрипты Perl внутри JVM без внешней версии Perl.

<https://code.google.com/p/jerl/>

Gnome 3.12

Рабочий стол под влиянием OSX улучшает отображение с высоким разрешением и преобразование приложений.

<https://gnome.org>

ZXTune b3000

Кросс-платформенный плейер «чиптюн» (имитация звука старых компьютеров) добавил поддержку SID-файлов C64.

<http://zxtune.bitbucket.org>

Horde Groupware Webmail Edition 5.1.4

Готовый для предприятия пакет коммуникаций на базе браузера.

<http://horde.org>

Qore 0.8.9

Язык скриптов с поддержкой многопоточности и встроенной логикой.

<http://qore.org>

Nuvie 0.5

Расширенный игровой движок для игр Origin: *Ultima 6*, *Martian Dreams* и *Savage Empire*.

<http://nuvie.sourceforge.net>



Игры старой школы Origin обрели новую жизнь в Linux.

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Улучшим

В этом месяце у нас на диске еще один пересобранный дистрибутив: на сей раз — смесь последних релизов Ubuntu, Kubuntu, Xubuntu и Lubuntu. Мы также добавили в процесс загрузки опцию, позволяющую выбрать из меню нужный вам рабочий стол. Ремиксы и модификации дистрибутивов вроде наших очень легко начать принимать как нечто само собой разумеющееся — не работу по изменению дистрибутива для DVD Linux Format, а сам факт, что мы можем это сделать. Тем, кто все время занимается программами с открытым кодом, легко забыть, какая это привилегия — иметь такую возможность.

Другие ОС не могут похвастаться подобной роскошью: их диски содержат лишь то, что им разрешили брать, и зачастую это урезанные или наспигованные рекламой версии программ. А мы можем не только включать ОС целиком, но и править их, чтобы DVD стал еще лучше, или менять ОС в наших мобильных устройствах. И это реальное преимущество. Нам стоит на минутку призадуматься о том, что это значит, и оценить колоссальные выгоды, которые дает открытый код, даже когда мы по лености предоставляем возиться с изменениями кому-нибудь другому.

Neil

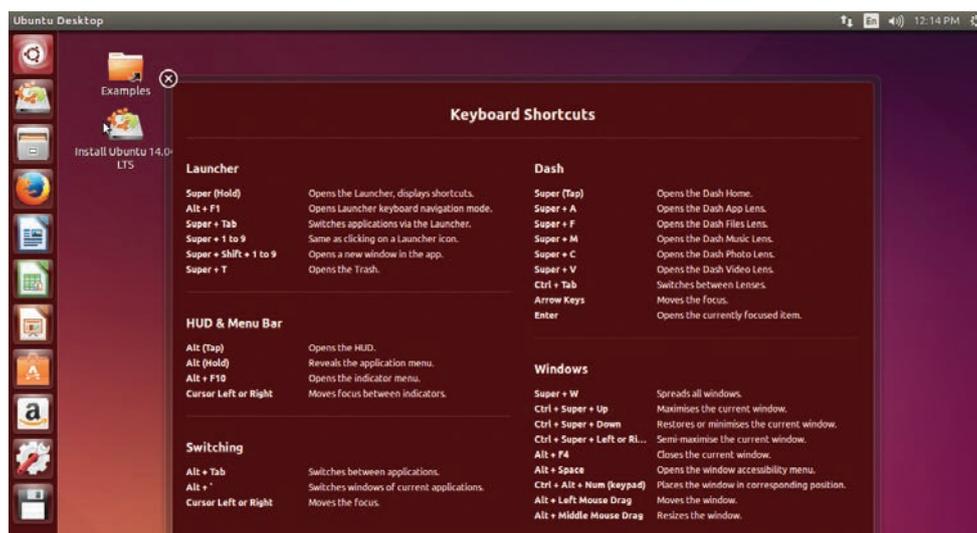
» Важно
ВНИМАНИЕ!

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

Ubuntu 14.04



Ubuntu 14.04 — это релиз LTS (Long Term Support, с долгосрочной поддержкой), и его поддержка теперь предлагается на пять лет. То есть в течение этого срока будут предоставляться исправления ошибок и обновления безопасности.

Как уже стало нормой, мы объединили вместе Ubuntu, Kubuntu, Xubuntu и новичка, Lubuntu. Все

они являются вполне официальными ответвлениями Ubuntu с альтернативными рабочими столами. Теперь познакомиться с ними стало намного проще: просто выбирайте тот, который вам приглянулся, в меню загрузки DVD. А если вам неохота перезагружаться, то старые добрые выход и вход тоже по-прежнему работают.

Дистрибутив Linux

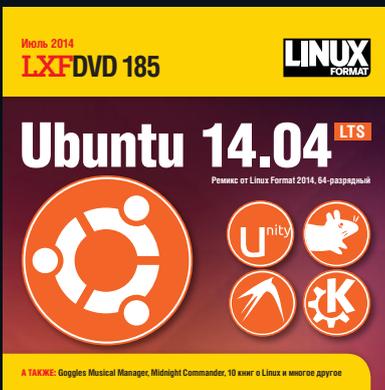
MX-14 “Symbiosis”

Мы привыкли видеть ответвления проектов и дистрибутивов, что не всегда к лучшему. Гораздо реже два проекта объединяются вновь, и именно это здесь и произошло. AntiX изначально основывался на Mepis, но с более легко-

весными рабочими столами для более старого оборудования, а потом стал прямым производным от Debian. Теперь две команды разработчиков трудятся совместно над релизом комбинированного дистрибутива — MX-14. Мы включили на DVD 32-битную версию, потому что этот дистрибутив идеально подходит для старого оборудования, а уж на более новом будет быстр невероятно.

Очень мало дистрибутивов свободны полностью. У большинства где-нибудь да есть кусок ограниченного кода; обычно это драйверы или прошивки или что-то от Adobe. Trisquel — один из немногих дистрибутивов, нацеленных на полную свободу: каждый байт кода свободен. Причем на нем все-таки можно смотреть видео с котятками на YouTube — похоже, это самая важная функция большинства ПК.





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

Дикие-дикие Wiki

Перепробуйте все самые странные в мире wiki из нашего Сравнения, включая: *Dokuwiki*, *Foswiki*, *MediaWiki*, *PmWiki*, *TiddlyWiki*.

Коды руководств

Программы и примеры для учебников этого месяца.



И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязательные быть в каждой операционной системе.

Hardinfo Инструмент для тестирования ОС.

Ядро Исходник последней версии ядра.

Memtest86+ Проверьте свое ОЗУ на сбои.

Plop Простой менеджер загрузки для запуска операционных систем.

RaWrite Создавайте загрузочные дискиеты в Windows.

SBM Независимый от ОС менеджер загрузки с простым интерфейсом.

Wvdial Соединяйтесь с Интернетом через телефонный модем.

Чтение

Книжная полка

The Cathedral and the Bazaar (Собор и Базар) Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

Intro to Linux Руководство для новичков.

Linux Kernel in a Nutshell (Ядро Linux в двух словах) Введение в ядро от магистра хакеров Грега Кроа-Хартмана [Greg Kroah-Hartman].

Справочник администратора Debian Руководство для системных администраторов.

Словарь Linux Linux от А до Z.

Окунитесь в Python Мастер-класс по этому популярному языку.

Руководство по созданию скриптов Bourne Shell Осваивайте скрипты оболочки.

Расширенное руководство по скриптам Bash Больше информации для бывалых.

Руководство сисадмина Основы работы с системой.



LXFHotPicks

Assassin of the Underdark

«Убийца в полумраке» — не сомнительный масонский титул, а исследование подземелий.

dispcalGUI

GUI утилиты калибровки цвета Argyll.

Git

Система управления версиями. По-английски — слово нехорошее.

GluCat

Библиотека моделирования шаблонных классов алгебр Клиффорда.

GogglesMM

Супер-быстрый менеджер фоно-теки и музыкальный плеер.

NuttX

Масштабируемая, нетребовательная к ресурсам RTOS.

Shellpic

Вывод изображений в терминал.

Qtractor

Многодорожечный аудиосеквенсор.

Pioneer

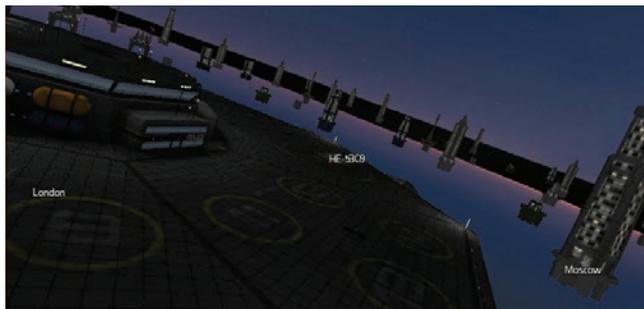
Игра, приключения в космосе.

Python

Свежая версия популярного языка.

Midnight Commander

Менеджер файлов... командир полуночи для убийц в полумраке?



Подкасты



Сезон 6 Эпизод 2

Хорошо, что в этом месяце он снова вышел, но мы надеемся дойти до третьего эпизода, если наконец сумеем изгнать маньяка-пианиста из студии звукозаписи.

Новости Red Hat объединяется с CentOS, блестящая Ye Olde SteamOS, менее блестящая Steam Machines, почему Steam VR становится пугающе хороша, хакеры, рассылающие спам, даты конца света для релизов Ubuntu.

Открытия недели Дистрибутивы для слабовидящих, распределенный доступ к данным NHS (отложенный) — хорошо или плохо? Изучение жесткого диска BlackBlaze с удивительными результатами.



Пропустили номер?



Закажите его через сайт www.linuxformat.ru в «ГНУ/Линуксцентре»! Журналы доставляются и в печатной, и в электронной форме, так что с момента открытия браузера до получения нужного вам выпуска LXF может пройти всего пара минут!

Прямо сейчас для заказа доступны следующие номера:



LXF182
Апрель 2014

250 руб.

- » **АНБ отдыхает** Не видать им наших данных
- » **Пускachi приложений** Моментальный залп
- » **Tor** Обдирать луковицу — наплачешься
- » **Сервер e-книг** Как пройти в библиотеку?

LXFDVD: Linux Tails, Netrunner, JonDo, Tor, TrueCrypt, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_182/
PDF-версия: shop.linuxformat.ru/elxf_182/



LXF183
Май 2014

250 руб.

- » **Битва дистрибутивов** Который лучше?
- » **Среда разработчика** Не просто редактор
- » **GnuPG** Модная тема криптографии
- » **Android + Linux** = Беседа телефона с компьютером

LXFDVD: Mageia 4, openSUSE 13.1, Ubuntu 13.10, LibreOffice 4.2, VMS Empire 1.11, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_183/
PDF-версия: shop.linuxformat.ru/elxf_183/



LXF184
Июнь 2014

250 руб.

- » **Стол поудобнее** Обустроим рабочее место
- » **Все под контролем** Мониторинг системы
- » **OpenELEC** Чисто конкретно для медиа
- » **Какой портрет, какой пейзаж!** Облагородим фотки

LXFDVD: Mint 16 Mega Mix, Ubuntu 14.04, Krita 2.8, The Legend of Edgar, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_184/
PDF-версия: shop.linuxformat.ru/elxf_184/

А чтобы не упустить ничего, оформите подписку! Все, кто подписался на печатную версию журнала через www.linuxformat.ru/subscribe/ или www.linuxcenter.ru, получают электронную версию в подарок! На сайте shop.linuxformat.ru вы также сможете приобрести предыдущие выпуски LXF.

Подписывайтесь на сайте www.linuxformat.ru/subscribe/

Телефоны отдела подписки:

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



Linux Format VKontakte

Вступайте в нашу
группу vk.com/linuxform

На странице LXF ВКонтате вы найдете:

- » Новости о Linux
- » Статьи из архива LXF
- » Анонс свежего выпуска LXF и часть статей из него
- » Живое общение и онлайн-консультацию по подписке на наше издание

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя_программы-1.0.1.tgz** — тот же файл, что и выше этажом по списку: «tgz» — это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

А ТАКЖЕ: SabotLinux 015 и Grated Live

» 64-разрядная сборка с рабочим столом Cinnamon
» 32-разрядная сборка с рабочим столом Mate

Mint 17



А ТАКЖЕ: Goggles Musical Manager, Midnight Commander, 10 книг о Linux и многое другое



Ubuntu 14.04

LTS

Ремикс от Linux Format 2014, 64-разрядный



Июль 2014
LXF DVD 185

LINUX
FORMAT

Июль 2014
LXF DVD 185

LINUX
FORMAT

Содержание

LINUX FORMAT

Сторона 1

ДИСТРИБУТИВЫ

Ubuntu 14.04 LTS С рабочими столами Unity, KDE, Xfce и LXDE, 64-разрядная версия
MX-14 Дистрибутив — гибридный дистрибутив Merp и Aptix, 32-разрядная версия
Trisquel 601 Полностью свободный дистрибутив
Gnml 2014.03 Дистрибутив на базе Debian

Сторона 2

ДИСТРИБУТИВЫ

Linux Mint 17 с рабочим столом Cinnamon, 64-разрядная сборка (ISO-образ)
Linux Mint 17 с рабочим столом Mate, 32-разрядная сборка (ISO-образ)
Gnbd.Linux 015 32-разрядная сборка (ISO-образ)
Grubred Live 32-разрядная сборка (ISO-образ)

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] «Собор и базар»

The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian

Dive Into Python Учебник по программированию на Python

Linux Dictionary Начальное руководство по Linux

Linux Dictionary Словарь Linux, объясняющий специфическую терминологию

Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Греггом Кроа-Хартманом [Greg Kroah-Hartman]

System Administrators Guide Руководство по базовому администрированию Linux.
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU.

NOTPICKS

Assassin Of The Underdark Игра, исследование подземелий

DispalcaGUI Графический интерфейс для утилит калибровки цвета

Git Система распределенного контроля версий

GitCat Библиотека классов шаблонов

GogglesMMI Легковесный менеджер медиа-файлов

Midnight Commander Файловый менеджер командной строки

MuttX Встраиваемая ОС

Pioneer Игра, космические приключения

Python Высокоуровневый язык программирования

Qtractor Многодорожечный секвенсор аудио/MIDI

Shelpic Просмотрщик изображений в терминале

Пожалуйста, перестаньте использовать названием диска озвучивать с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аплиментарный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. Если вы обнаружили дефект на диске, пожалуйста, сообщите об этом в редакцию Linux Format не в состоянии принести на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут возникнуть в результате использования этого DVD, предоставленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с сайта www.linuxformat.ru и скачайте программу для загрузки с сайта www.linuxformat.ru.

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 GNU/LINUX НА USB FLASH



Linux Mint 16



Ubuntu 14.04

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 № ФФ77-21973 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

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

ЗАО «Мезон.Ру»

Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. б/н, лит. Ф
Тел. (812) 462-8383, e-mail: office@ldprint.ru
Заказ 10973

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор (Neil Mohr) neil.mohr@futurenet.com

Заместитель редактора Мэтью Хэнсон (Matthew Hanson)

matthew.hanson@futurenet.com

Научный редактор Джонни Бидвелл (Jonni Bidwell) jonni.bidwell@futurenet.com

Выпускающий редактор Крис Торнетт (Chris Thornett)

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоза

efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Джонни Бидвелл (Jonni Bidwell), Нейл Ботвик (Neil Bothwick), Крис Браун (Chris Brown), Джоллон Браун (Jolyn Brown), Кент Ельчук (Kent Elchuk), Дункан Эванс (Duncan Evans), Марко Фиоретти (Marco Fioretti), Мэтью Хэнсон (Matthew Hanson), Дэвид Хейвард (David Hayward), Фил Иванюк (Phil Iwanuk), Робин де Йонг (Robin de Jongh), Нейл Мор (Neil Mohr), Лес Паундер (Les Pounder), Даниэль Самюэлс (Daniel Samuels), Маянк Шарма (Mayank Sharma), Ричард Смедли (Richard Smedley), Евгений Балдин, Александр Толстой, Алексей Федорчук, Ирина Чернова, Игорь Штомпель

Художественный ассистент Ник Эспелл (Nick Aspell)

Иллюстрации Шейн Коллиндж (Shane Collinge), Саймон Миддлвик (Simon Middleweek), iStock Photo

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 01225 442244, email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):

пр. Медиков, 5, корп. 7. Тел. +7 (812) 309-0686

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел./факс +7 (499) 271-4954

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вам 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 повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса (Linus Torvalds). «GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>.



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

В августовском номере

Как настроить Ubuntu 14.04

Trusty Tahr явился — но стоит ли принимать его как таковой? Мы покажем, как настроить, подправить и улучшить свежий релиз, сделав его идеалом для ваших нужд.

При малых ресурсах

Какой дистрибутив удобнее всего усядется на старом оборудовании, вернув его к жизни? Разберемся.

Менеджеры пакетов

Руководство по менеджерам пакетов и их применению — на свете есть не только *apt-get*!

Пройти экзамен LPI

Выберите верный способ получить сертификат Института профессионалов Linux, по личным либо карьерным причинам.

Содержание будущих выпусков может меняться — вдруг нас приструнит Canonical...

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)

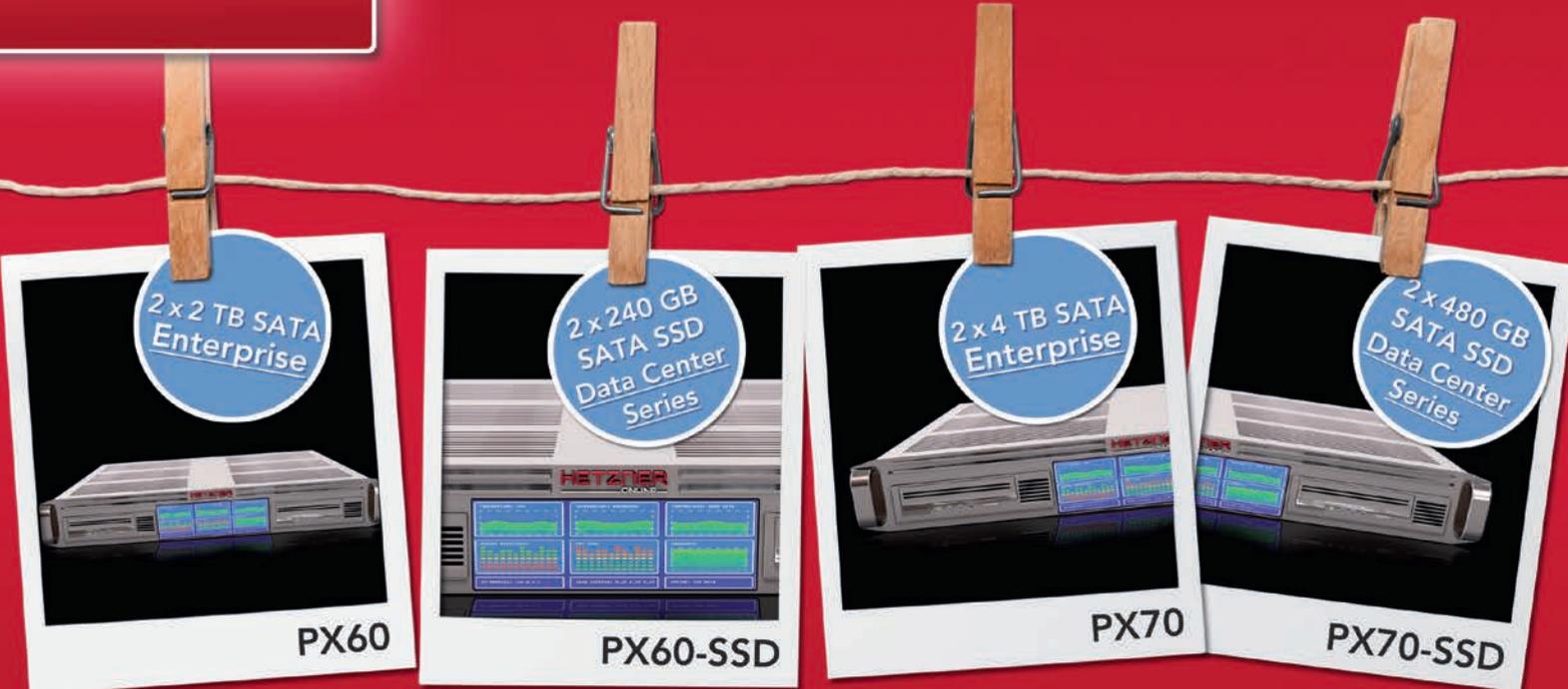
309-06-86

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

Linux  center

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 ГБ 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 ГБ 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.