

DVD ДАРОМ!



ПЛЮС ЦЕННАЯ e-КНИГА

LINUX FORMAT

Главное в мире Linux

46 страниц
проектов
& Raspberry Pi

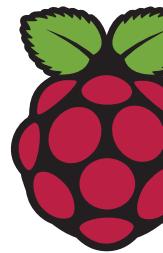
Бронезащита

» За вами следят?

Строим SquirrelMail

» Ваша личная web-почта

ПЛЮС 3D-графика OpenGL
на Raspberry Pi

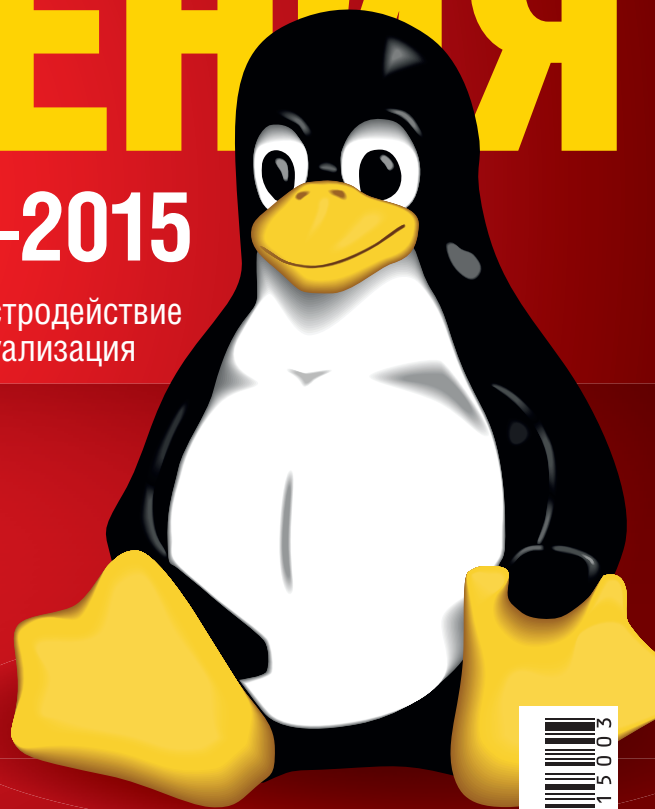


Март 2015 № 3 (194)

LINUX НОВОГО ПОКОЛЕНИЯ

Самые крутые функции-2015

» Лучший рабочий стол » Графика Wayland » Отличное быстродействие
» Улучшенная файловая система » Новейшие ядра » Виртуализация



Академия КОДИНГА

**8 СТРАНИЦ ПРОГРАММНОГО
СОВЕРШЕНСТВА!**

Java Создаем простую web-игру **с. 88**

Erlang Вникните в этот язык **с. 92**



Докинг завершен

« IBM обнаружила,
что Docker в 26 раз
быстрее виртуальной машины »

Джеймс Тернбалл — про революцию Docker **с. 38**



Firefox 10 лет!

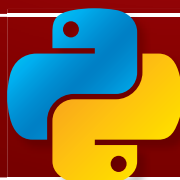
» Как открытый
браузер побивает
Microsoft



Сравнение

Рабочие столы Linux

» Найдите рабочую среду
своей мечты



Arduino

Micro Python

» Создаем встраиваемые
проекты реального времени

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

Linux center
www.linuxcenter.ru



iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ

Что мы делаем

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



Кто мы

На сей раз мы спросили наших экспертов: что они ожидают/надеются/хотят увидеть в GNU/Linux 2025 года?



Джонни Бидвелл

Как насчет единой системы init/оболочки/рабочего стола, настолько классно настроенных и надежных, что все единодушно хотят их использовать? И ПО пускай все принудительно скачивают через киоск приложений. Да чтобы «железо» тоже не отставало!.. Вообще-то ничего хорошего тут нет. Я надеюсь, что в 2025-м все еще буду сражаться с config'ами и командной строкой.



Шон Конвэй

Я надеюсь, что за следующие 11 лет GNU/Linux доберется до настольных ПК Северной Америки. Для этого всего лишь требуется история успеха с подробностями о резком снижении расходов за счет отказа от проприетарного ПО, тогда больше компаний начнут просчитывать шансы сэкономить. Лично у меня будет какая-нибудь Fedora 32.



Лес Паундер

В 2025 году повсеместной станет переносная технология, а после выпуска Raspberry Pi модели G+ дети смогут перепрограммировать друг друга в классе благодаря GPIO, прикрепленному к позвоночнику. Учителя будут загружать уроки прямо в мозг школьников и собирать домашние задания через Интернет, используя хэш MD5, чтобы убедиться в подлинности автора.



Маянк Шарма

Было бы здорово, если бы в GNU/Linux стало меньше агитации и больше прагматизма. К 2025 году разработчики открытого и свободного ПО должны уже придумать, как снабдить дистрибутивы роботом-вантузом. А пользователи потом смогут с его помощью вычистить проприетарное ПО из своего компьютера и наслаждаться GNU/Linux.



Валентин Синеицын

2025? Разве это не год Linux на рабочем столе? В первый раз я услышал эту фразу как раз лет 10 назад, но кто тогда знал, что в 2015-м рабочий стол уже не будет столь актуален? Думаю, к 2025 году Linux будет комфортно работать на каких-нибудь новых гаджетах, без которых мы не будем мыслить себя.



Уходящая натура

» В кино- и фотодокументалистике этим термином называют то, что скоро невозможно будет снять. То, что безвозвратно уходит в прошлое. Когда компания Microsoft объявила, что обновление до Windows 10 станет бесплатным даже для нелегальных пользователей, в этот разряд попало столь привычное явление, как платная ОС для рабочих станций.

Ранее то же самое случилось с проприетарными UNIX для платформы x86: они уступили «экологическую нишу» свободным системам, в первую очередь — Linux.

Стратегия бизнеса обоих изготовителей проприетарных ОС для десктопов — Apple и Microsoft — теперь одинакова. Прибыль приносят облачные сервисы и «железо», а всё остальное можно просто подарить. Как и в случае с проприетарным UNIX, свободное ПО тоже сыграло свою роль. Объяснить пользователям (особенно корпоративным, где количество рабочих станций измеряется тысячами), за что у них просят деньги, стало несколько затруднительно.

Похоже, что в обозримом будущем и само «абонентское устройство» (компьютер, планшет или еще что-нибудь) можно будет получить бесплатно при покупке годовой подписки на ту или иную облачную услугу. Операторы сотовой связи уже такое умеют.

Платные настольные ОС уходят. Не забудьте сделать снимок экрана!

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

Знаете всё о Linux?

ДА

НЕТ

Станьте автором
в журнале Linux Format!
linuxformat.ru/avtoram.phtml

Зарегистрируйтесь
на сайте
shop.linuxformat.ru



СОМНЕНИЯ

Скачайте бесплатно
архивные PDF-номера
журнала с сайта
linuxformat.ru/archive
и загляните на
wiki.linuxformat.ru



Версия для iPad
и iPhone доступна
в App Store

Выберите вид подписки

PDF-версия
на 6 месяцев

990 ₺

PDF-версия
на 12 месяцев

1800 ₺

Печатная версия
на 6 месяцев

1890 ₺

Печатная версия
на 12 месяцев

3480 ₺

Выберите вид доставки

Оплатите

Читайте Linux Format!

Станьте Linux-гуру

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

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

Печатная версия Linux Format понравится читателям, которые:

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

Способы доставки

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

Способы оплаты

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



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

Содержание

«Стало до ужаса ясно, что наша технология опередила нашу гуманность.»

Обзоры

AV Linux 6.0.4 14

Дистрибутив Linux напихивали инструментами редактирования видео и аудио.

The SSS 10.1 15

Серверных комплектов меньшего размера просто не бывает.

Toshiba Chromebook 2 16

Наросло новое поколение хромбуков, и пользоваться ими — истинное удовольствие.

Intel Core i7-5820K 17

Больше не ищите: мы объясним, почему надо скорей хватать этот 6-ядерный процессор.

Samsung 845DC 18

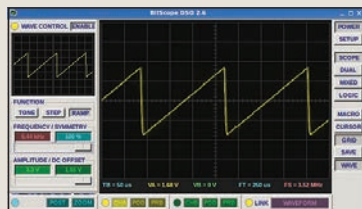
Наконец-то появился твердотельный диск, способный выдюжить серверную нагрузку. Цена, правда, кусается.



Этот SSD-долгожитель посрамил зайчика Duracell.

BitScore BS05 19

Оциллограф для Raspberry Pi? Почему бы и нет? И управление как у больших.



Вы найдете в BitScore DSO всё, что полагается.

Lumino от 4tronix 20

Недорогая микро-робоплатформа на Arduino.

Geometry Wars 3 21

Ни за что не игайте! Иначе — лютая смерть. Стрелялки, они такие...

Пулеупорные SSL и TLS 21

Книга для сисадминов о том, как держать оборону, с цитатами из Сноудена.

ТЕХНОЛОГИИ

2015



Технологии Linux нового поколения 2015 года доступны уже сейчас! с. 30

Сравнение: Рабочие столы с. 24



Люди говорят



Oracle жаждет стать клиентом нашей компании со штатом в 45 сотрудников. Потрясающе!

Джеймс Тернбалл — о шквальном успехе Docker с. 38

На вашем бесплатном DVD

Побалуйте себя и любимых подпиской на LXF!

LINUX ЛУЧШИЕ ДИСТРИБУТИВЫ И БОЛЕЕ ТОГО! ВСЕ ДЛЯ БЫСТРОГО СТАРТА В LINUX

FEDORA 21

РАБОЧАЯ СТАНЦИЯ
Главные авангардные функции Red Hat



GIMP
PHOTO EDITING MADE EASY
GIMP: Редактируем фото запростой!
БЕСПЛАТНАЯ Е-КНИГА

Manjaro
Мощный дистрибутив, популярный каждому!

Fedora 21 Workstation, Manjaro 8, 4M Linux, ALT Linux Enlightenment

» Убойные дистрибутивы по теме номера

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



Доступно в AppStore!



www.linuxformat.ru/subscribe

Ищите в этом номере

Firefox'у — десять лет! 44

Ну вот как в Mozilla сумели сделать мировой браузер?

Raspberry Pi и 3D на OpenGL 48

Ускорение OpenGL ES стало проще с Pi3D на Python.

Пренатальная Betsy 52

Делаем УЗИ еще не выпущенному релизу Linux Mint на базе Debian.

Matuntu как он есть 60

Никакой ненормативной лексики: Ubuntu на языке родных осин.



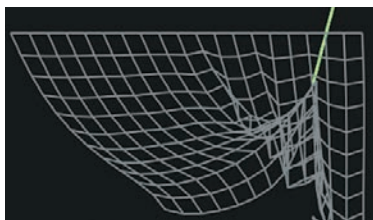
Академия коднга

Физика на Java 88

Разберемся, как пользоваться Matter.js — физическим движком на JavaScript. Хотя он и невелик, но для двумерных игр его мощности хватит с избытком.

Erlang 92

Введение в язык: сперва забудьте всё, что знали. Мы тут про функциональное программирование толкуем! От сбоев системы никуда не денешься — ну так по крайней мере их можно засечь и пресечь.



Постоянные рубрики

Новости 6

Выпала очередная Роса, *ownCloud* выдвигает концепцию, ядро починается на ходу, Ubuntu и Tizen проникают в телефоны, распространяются облачные сервисы и вирусы, а Gnome проявляет заботу о меньшинствах.

Новости Android 22

Зеленые человечки закиданы яблоками, Microsoft вложился в Суапоген, Китай шагнул в Америку, а умным часам недостает времени.

Сравнение 24

Надоел свой рабочий стол? Помешайте! Мы потерзали пятерку лучших в нашей тестовой лаборатории. Cinnamon, Enlightenment, Gnome Shell, KDE Plasma, Mate

Интервью LXF 38

Джеймс Тернбалл недоволен качеством документации к открытому коду и в восторге от успехов Docker.

Рубрика сисадмина 62

Доктор ударился в поэзию, а потом в пакетную обработку отпускных фотографий. И наконец-то завершил свою сагу об OpenLDAP. Что не может не радовать.

Ответы 96

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик — про синхронизацию с Google Drive, копирование LXF DVD, простое решение для удаленного доступа, медленное копирование с *scp*, утерянный пароль для входа в систему.

HotPicks 100

Отдайте горяченького! Лучшие в мире новинки свободного ПО: *Audex, Budgie Desktop, Chocolate Doom, ClamAV, Debreate, Fifth, GNU Hello, QuiteRSS, VoltAir, XDecorations, XFDesktop*.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

Придется раздобывать, в противном случае в будущем вам придется туго.

Через месяц 112

Откройте секреты следующего номера — да не проболтайте конкурентам, а то наши идеи сопрут!

Учебники

Безопасность Осторожно: вторжение! 68

Изучим, как оснастить безопасный порт сервиса и нарочно оголить его перед Интернетом, чтобы узнать: а не подглядывают ли за вами какие-нибудь Большие Братья?

DTTrace Мониторинг системы 70

Всё, что вам следует знать об удобном инструменте *DTTrace*, чтобы проверить производительность своей системы. Воткнем зонды повсюду!

Gummiboot Шустрый загрузчик 74

Настало время стащить *Grub* на свалку и обратиться к более быстрому варианту, который ладит с UEFI.

Squirrel Mail Личный сервер web-почты 76

Как настроить и запустить свой собственный почтовый web-сервис, безопасности ради и развлечения для.

Micro Python Встроенные контроллеры 80

Комбинация из трех... компонентов — Python, Pi и Arduino — позволит создать аппаратные решения реального времени.

ПЛИС Что-то с памятью 84

Вдоволь намигавшись светодиодами, мы придумались: а для задач посложнее, не маловато ли в нашем процессоре ОЗУ? Впрочем, это дело поправимое.



В ЭТОМ НОМЕРЕ: Новый русский дистрибутив » Идея ownCloud » Горячая замена ядра
» Ubuntu в телефоне » Смартфон с Tizen » На OpenStack » Вирус из Китая » Gnome и ЛГБТ

ЗИМНЯЯ ROSA

На все случаи жизни

Вышел новый дистрибутив с легковесным рабочим столом LXDE.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

Начало 2015 года российская компания «НТЦ ИТ РОСА» отметила выпуском свежей редакции дистрибутива ROSA Desktop Fresh R5 с рабочим столом LXDE. Дистрибутив нетребователен к видеокарте, а интерфейс системы схож с Windows XP. R5 подойдет и для работы, и для досуга, если вы не намерены изучать новые концепции интерфейсов или глубоко настраивать систему «под себя», и облегчит пользователям процесс импортозамещения, особенно тем, кто «переезжает» с Windows 95/XP/7 и привык к классическому рабочему столу с нижней панелью. ROSA Desktop Fresh R5 корректно работает на «слабых» конфигурациях, от старых ПК и ноутбуков до современных экономичных нетбуков.

В репозиториях пользователей ждут десятки тысяч собранных свободных программ и самые распространенные кросс-платформенные продукты, такие как Skype, Dropbox, Steam, плюс большая кол-

«ROSA Desktop Fresh LXDE R5 подойдет и для работы, и для досуга.»

лекция игр и эмуляторов. Работает просмотр видео разных видов и форматов. Ну и «свежий» LibreOffice, полный TeX-стек и широкий набор программ для документирования, верстки и полиграфии. Для разработчиков — LAMP/Ruby/Node/C++/...

Часть дополнений разработки «НТЦ ИТ РОСА» касается эргономики: реализованы привычная для пользователя «точка входа» для настройки системы и «центр управления». Появились загрузка в SecureBoot-режиме и много улучшений загрузчика; обновлены версии ядра, библиотеки MESA и X11, графические драйверы NVidia и AMD и прикладное ПО; раз в неделю автоматически запускается TRIM для SSD, пакет *ssd-utils* включен в дистрибутив; введены поддержка систем с гибридной графикой Intel+AMD, новые версии XFdrake, индикатор батареи и управление питанием из Xfce; исправлен ряд ошибок (в т.ч. баг ROMP с воспроизведением файлов с именами на кириллице). Срок поддержки этой версии — 2 года.

ЗА ОБЛАКАМИ

ownCloud Server 8

Новая концепция меняет подход к совместному использованию данных.

Новые возможности, реализованные в ownCloud Server 8, ускоряют обзор и совместное использование хранящегося в «облаке» контента. Обновленный интерфейс поиска предоставляет расширенный доступ к дополнительным параметрам и ускоряет доступ к найденным файлам. Можно сделать Избранными чаще всего используемые файлы и папки. Ранее совместный доступ к файлам в ownCloud осуществлялся через генерацию ссылок, которыми владелец делился с другими пользователями. Такое практикуют многие решения облачного хостинга, скажем, DropBox и SpiderOak.

Концепция ownCloud Server 8 — Federated Cloud Sharing — позволяет не создавать вручную ссылки для каждого файла, а обратиться к целевым пользователям других облачных серверов через share-диалог, уведомляющий, что файл предоставлен в доступ. Франк Карличек [Frank Karlitschek], создатель ownCloud, заявляет: «Эта концепция в корне меняет

«Демо-приложение пригодно как основа при написании программ.»

представление о безопасности при совместной работе. А нововведения вроде Избранного и улучшения поиска — основа будущей функциональности метаданных в ownCloud, которая окажет значительное влияние на опыт взаимодействия. Для администраторов улучшены производительность и элементы управления».

Разработчикам рекомендуется перечислять зависимости своих приложений в файле *info.xml* — контейнер приложений ownCloud автоматически их разрешит. Новичкам поможет новое руководство с демо-приложением, пригодное как основа при написании программ для ownCloud.

МОБИЛЬНАЯ СТРАТЕГИЯ

Ubuntu здесь, Ubuntu там...

Появился первый телефон на «чистом» Linux.

Для работы с данными, хранящимися онлайн, лучшим приложением Canonical считает браузер.

В феврале в Европе начались продажи телефона, который с полным правом может быть назван «Ubuntu Phone»: малоизвестный испанский производитель BQ совместно с Canonical представил на рынок модель Aquarius E4.5 Ubuntu Edition.

Аппаратная составляющая новинки соответствует Android-версии аналогичного смартфона BQ: экран с разрешением 540×960, 4-ядерный процессор MediaTek Cortex A7 1,3 ГГц, 1 ГБ ОЗУ и 8 ГБ встроенной памяти. Устройство поддерживает внешние карты памяти microSD и оснащено двумя камерами: основной, с разрешением 8 МПикс, и фронтальной — 5 МПикс. По нынешним временам, спецификации достаточно скромные. Тем не менее, выпуск Aquarius E4.5 стал историческим моментом не только для Canonical, но и для всего сообщества Linux: это первое в мире устройство с предустановленной Ubuntu Phone OS — «чистым» GNU/Linux, полностью открытой ОС. Все другие

мобильные операционные системы на основе Linux, в отличие от Ubuntu Phone, либо имеют не полностью открытый исходный код, либо на них не могут работать приложения для Linux.

Да, традиционные приложения Linux, создаваемые для рабочих станций, нет смысла запускать на 5-дюймовом экране мобильного устройства. Но разработчикам, например, *Gedit* или *GIMP* теперь будет гораздо проще перенести свои приложения на мобильную платформу. Пока же Canonical демонстрирует совершенно иную стратегию: традиционные приложения имеют смысл, когда обработка и хранение данных происходят локально. Для доступа же к сайтам и онлайн-сервисам лучшее приложение — браузер!

Ubuntu Phone использует «умные» области *Aggregator Scores*, которые, подобно *Google Now*, при помощи *Web API* собирают соответствующую информацию из различных источников и предоставляют пользователю результат. В отличие от Android или iOS, домашние экраны Ubuntu Phone разделены по категориям (*Music*, *Movies*

» «Убунтофон» Aquaris E4.5 Ubuntu Edition — первый смартфон с полноценным Linux на борту.



и т. д.), на которых группируется контент с разных сайтов.

Новинка работает не только с web-приложениями: сообщество Ubuntu активно переносит нативные приложения на мобильную платформу. Для Ubuntu Phone уже доступен, в частности, даже терминал.

«Убунтофон» предлагается по цене €169,90 (около \$200) и будет продаваться только в Европе.

ЯДРО LINUX

Чисть машину на ходу

В ядре Linux 4.0 появляется поддержка технологии Live Kernel Patching.

В течение прошлого года компании SUSE и Red Hat разработали решения для реализации технологии live kernel patching — применения патчей к работающему ядру без перезагрузки системы, названные *kGraft* и *Kratch* соответственно. Компании вели разработку параллельно, независимо друг от друга, однако после обнародования *kGraft* и *Kratch* пришли к соглашению о сотрудничестве, направленном на создание общей базы, которая адресует потребности каждой реализации.

В результате этого сотрудничества был предложен универсальный API, не привязанный к конкретной реализации. Совместная разработка SUSE и Red Hat позволит применять исправления к работающему ядру «на лету», не требуя перезагрузки системы.

Первым выпуском ядра, содержащим такое комбинированное решение, будет Linux kernel 4.0.

Пользователи GNU/Linux получили новый аргумент для привлечения потенциальных неопитов, переходящих с Windows или Mac OS — теперь обновленная система GNU/Linux не нуждается в перезагрузке в случаях, когда прежде это было необходимо: после установки драйверов устройств или обновления ядра. Технология live kernel patching позволит конечным пользователям и системным

«SUSE и Red Hat пришли к соглашению о сотрудничестве.»

администраторам не перезагружать свои работающие системы в течение нескольких месяцев подряд, своевременно обновляя при этом ядро.

Список рассылки сообщает, что пока реализована поддержка только архитектуры x86 и продолжаются работы для PowerPC, s390 и ARM; однако насчёт x86-64 не сказано ни слова. Поясняется, что после объединения общей инфраструктуры Red Hat и SUSE согласились сразу начать портировать свои текущие решения, отказавшись от специфичного для компаний кода. План, в основном, заключается в том, что каждый патч будет отмечен флагом, указывающим, какую модель непротиворечивости он готов использовать.

Линус Торвалдс уже принял код базового API для обновления без перезагрузки в состав ядра Linux 4.0.

НЕ ТОЛЬКО ANDROID

Даешь мобильные платформы!

Samsung выпустила свой первый смартфон на платформе Tizen.

Samsung Electronics начала в Индии продажи своего первого смартфона на платформе Tizen. Модель назвали Z1; стоит она 5700 индийских рупий (\$92).

Samsung представила серию смартфонов Z под управлением открытой ОС Tizen, в июне 2014 г. на Tizen Developer Conference в Сан-Франциско. Первые продажи намечались в России (в III квартале 2014 г.), но так не вышло. Да и продажи в Индии опоздали к плановым срокам.

От смартфона Z модель Z1 отличается относительно низкими спецификациями. Так, Z комплектуется 4,8-дюймовым дисплеем HD Super AMOLED и 4-ядерным процессором 2,3 ГГц. Z1 работает под управлением Tizen 2.3, оснащён 4-дюймовым дисплеем с разрешением 480×800, двухядерным процессором 1,2 ГГц, 768 МБ ОЗУ и 4 ГБ внешней памяти microSD (расширяемой до 64 ГБ). Аккумулятор ёмкостью 1500 мА·ч обеспечивает 7 часов воспроизведения видео или 8 часов разговоров.

Итак, Samsung проявила интерес к нижнему ценовому сектору рынка, быстро растущему в Индии и других развивающихся странах. «В Индии Z1 будет конкурентом устройств на Android, с похожими спецификациями и ценами, — говорит аналитик компании Gartner Вишал Трипатхи [Vishal Tripathi]. — Для роста популярности смартфонам на Tizen нужно время: многие пользователи ещё не знакомы с этой системой. Мотивацией для смены платформы могут стать предлагаемые Tizen приложения, особенно если они превзойдут аналоги для Android. Но пока разработчиков для Tizen недостаточно».

Предустановленные на Z1 приложения явно показывают, что основное назначение этого устройства — развлечения.

Samsung на 3 месяца выдаёт покупателям Z1 бесплатный доступ к фильмам и музыке на популярных индийских развлекательных сайтах, а также к премиум-контенту на собственном развлекательном портале Club Samsung. Местные провайдеры услуг мобильной связи Reliance Communications и Aircel 6 месяцев обеспечивают пользователю Z1 бесплатную передачу по 500 МБ 3G-данных ежемесячно.

Ещё одна «фишка» новинки — умение подать сигнал SOS. 4-кратное нажатие на кнопку выключения питания Z1 отправит сообщение Help по основным контактам пользователя, указав, где он.

» Samsung Z1 под управлением Tizen хорош, но большинство пользователей пока ничего не знают об этой платформе.



ВОЗЬМЕМСЯ ЗА РУКИ, ДРУЗЬЯ

Облака на российском рынке

«Сервионика» и «Мирантис» объединяют усилия по разработке на OpenStack.

Сотрудничество позволит компаниям объединить успешный опыт разработки, внедрения и эксплуатации облачных сервисов. Российский провайдер облачных сервисов компания «Сервионика» (ГК «Ай-Теко») и занимающаяся разработкой и поддержкой OpenStack компания «Мирантис» начинают реализацию совместных проектов на базе этой облачной платформы — популярной и активно развивающейся.

«Мирантис» и «Сервионика» считают перспективным совместное продвижение OpenStack на российском рынке, где все более востребовано ПО от отечественных разработчиков на базе открытого кода. «Мирантис» разработала один из наиболее успешных коммерческих дистрибутивов OpenStack, представленный на глобальном рынке, а «Сервионика» — первый российский сервис-провайдер с реальным опытом коммерческой эксплуатации облачного сервиса на OpenStack.

В 2012 г. «Сервионика» вывела на рынок сервис MakeCloud — «облачный

конструктор» с универсальным набором функций для самостоятельного создания и управления виртуальной инфраструктурой. В 2014 году MakeCloud лёг в основу нового решения «Сервионика» — Ru-Stack, для крупных корпоративных

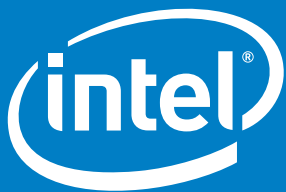
«Разработкой этого проекта занимаются более 180 компаний.»

клиентов. Ru-Stack позволяет управлять виртуальными ресурсами на разных платформах виртуализации (VMware, KVM, Hyper-V и других), как для реализации внутренних проектов крупного бизнеса, так и для предоставления внешних коммерческих облачных сервисов. Оно объединяет возможности открытых и проприетарных технологий: высокую надежность и масштабируемость, простоту развертывания, оптимальную стоимость и сокращение издержек на развитие.

Компании уже начали работу над совместным демо-стендом на Mirantis OpenStack, позволяющим увидеть в действии функционал создания и управления виртуальной ИТ-инфраструктурой для различных задач. OpenStack — облачная платформа с открытым кодом, распространяемая под лицензией Apache 2.0. Разработкой этого проекта, позволяющего создавать облачную инфраструктуру и хранилища данных, занимаются уже более 180 компаний из разных стран мира.

» Рост популярности решений на базе открытого кода в России позволяет реализовать масштабные и сложные проекты.

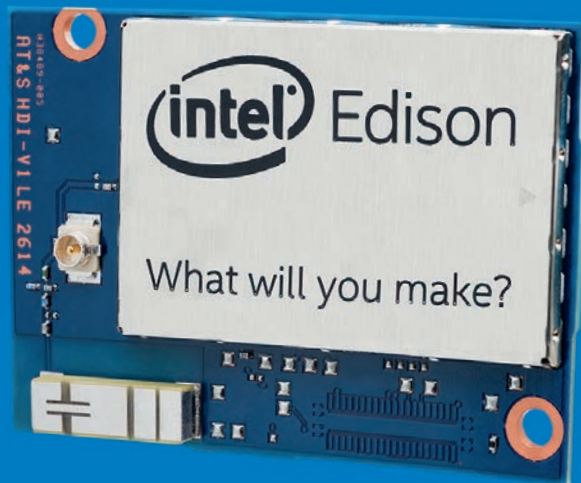




Intel® Edison

Впервые в России [официально].

Оформи предзаказ на
www.linuxcenter.ru



Выиграй поездку*
на Geek Picnic
[19–21 июня] или
Intel IoT Roadshow
[27–28 июня]
в Санкт-Петербурге.

Москва: +7 (499) 271-49-54
Санкт-Петербург: +7 (812) 309-06-86
E-mail: info@linuxcenter.ru



С полными условиями конкурса
можно ознакомиться на
www.linuxcenter.ru/intel/edison



* в призовом фонде 6 поездок

ВИРУСЫ

Под чужой командой

Новый бэкдор для Linux весьма многофункционален.

У вирусных аналитиков компании «Доктор Веб» имеются основания полагать, что новый бэкдор создала китайская хакерская группа ChinaZ.

Специалисты компании исследовали многофункционального троянца, заражающего ОС Linux и умеющего выполнять команды злоумышленников, в т.ч. организовывать DDoS-атаки. Новый троян, названный *Linux.BackDoor.Xnote.1*, распространяется аналогично своим сородичам: подбирая пароль, злоумышленники взламывают учетные записи для доступа к атакуемой системе по протоколу SSH.

Установка трояна в систему возможна только от имени суперпользователя (root): при этом троян создает свою копию в папке */bin/* в виде файла с именем *iptables6* и удаляет исходный файл, из которого был запущен. Также *Linux.BackDoor.Xnote.1* ищет в папке */etc/init.d/* скрипты, начинающиеся со строки *#!/bin/bash*, и добавляет

после этой строки другую, обеспечивающую запуск бэкдора. По команде извне *Linux.BackDoor.Xnote.1* может выдать зараженной машине уникальный идентификатор, начать DDoS-атаку на удаленный узел с заданным адресом (среди возможных типов атак — SYN Flood, UDP Flood, HTTP Flood и NTP Amplification), остановить атаку, обновить свой исполняемый файл, записать информацию в файл или удалить себя. Он отправляет «хозяевам» данные о файловой системе инфицированного компьютера (общее количество блоков данных, количество свободных блоков), может принять файл, отправить файл на управляющий сервер, удалить файл или каталог, переименовать или запустить файл или командную оболочку [shell] со своими переменными окружения, выдав управляющему серверу доступ к ней, запустить на компьютере-жертве SOCKS проxy или свою версию сервера portmap.

GNOME РАД ВСЕМ

Смена вывески

Gnome-инициатива OPW переименована в Outreachy.

Проводимая под эгидой Gnome Outreach Program for Women (OPW, Программа привлечения женщин) имела целью увеличить присутствие представительниц прекрасного пола в проектах открытого ПО, будь то сотрудничество в программировании или же участие в решении связанных задач — работа над документацией, дизайном и т.д. Участвуя в деятельности OPW, дамы приобретали опыт и получали вознаграждение за работу. Так, тесные связи с OPW имел основанный нынешней зимой проект X.Org Foundation.

Отныне OPW переименована в Outreachy, а новая цель инициативы — «обеспечение стажировки в проектах свободного и открытого ПО людей из групп, в силу гендерной самоидентификации слабо представленных в открытом сообществе», то есть не только женщин. Желающим поддержать сообщество предлагают сперва внести свой вклад в любой момент

в течение года. 2 раза в год предоставляется целенаправленная стажировка в организациях, создающих свободное ПО. Предстоящий раунд стажировок открыт для транссексуалов, гендерквиров и всех участников проекта Ascend Project независимо от пола. Планируется далее расширить программу на недостаточно представленные в сообществе меньшинства.

Поддержку программе оказывают такие организации, как Red Hat, Software Freedom Conservancy и др. Подробности — на сайте gnome.org/outreachy. LXF



➤ Проект Outreachy привлекает в сообщество Open Source не только женщин.

Новости короткой строкой

➤ Руководитель думского Комитета по информационной политике, технологиям и связи Леонид Левин предложил ограничить доступ к анонимным средствам доступа в Интернет (сеть Tor и т.п.), хотя готовых законопроектов, реализующих эту идею, пока нет. Источник: www.vedomosti.ru

➤ Фонд СПО включил Guix в список полностью свободных дистрибутивов, а лицензии Creative Commons Attribution 4.0 International и Attribution-ShareAlike 4.0 International — в список свободных лицензий, не предназначенных для ПО и документации. Источник: www.fsf.org

➤ Столлмен отвергает добавление в Emacs отладчика LLDB помимо штатного GDB, считая это «попыткой систематических атак на пакеты GNU». Источник: www.reddit.com

➤ Очередной выпуск системной библиотеки GNU C Library (*glibc*) 2.21 полностью соответствует стандартам ISO C11 и POSIX.1-2008. Источник: sourceware.org

➤ Филип Ньюборо [Philip Newborough], автор и ведущий разработчик CrunchBang Linux, оставил проект, посоветовав сообществу перейти на Debian. Источник: crunchbang.org

➤ Объединённая команда разработчиков LXDE, Razor-qt и Maui/Hawaii выпустила второй знаковый релиз нового рабочего окружения — LXQt 0.9. Источник: lxqt.org

➤ Microsoft переводит серверную платформу .Net Core в разряд открытого ПО и публикует код движка .NET-приложений CoreCLR под лицензией MIT. Источник: blogs.msdn.com

➤ Проект Debian завел новое зеркало в Японии, security.debian.org, предоставленное SAKURA Internet, Inc. Источник: www.debian.org

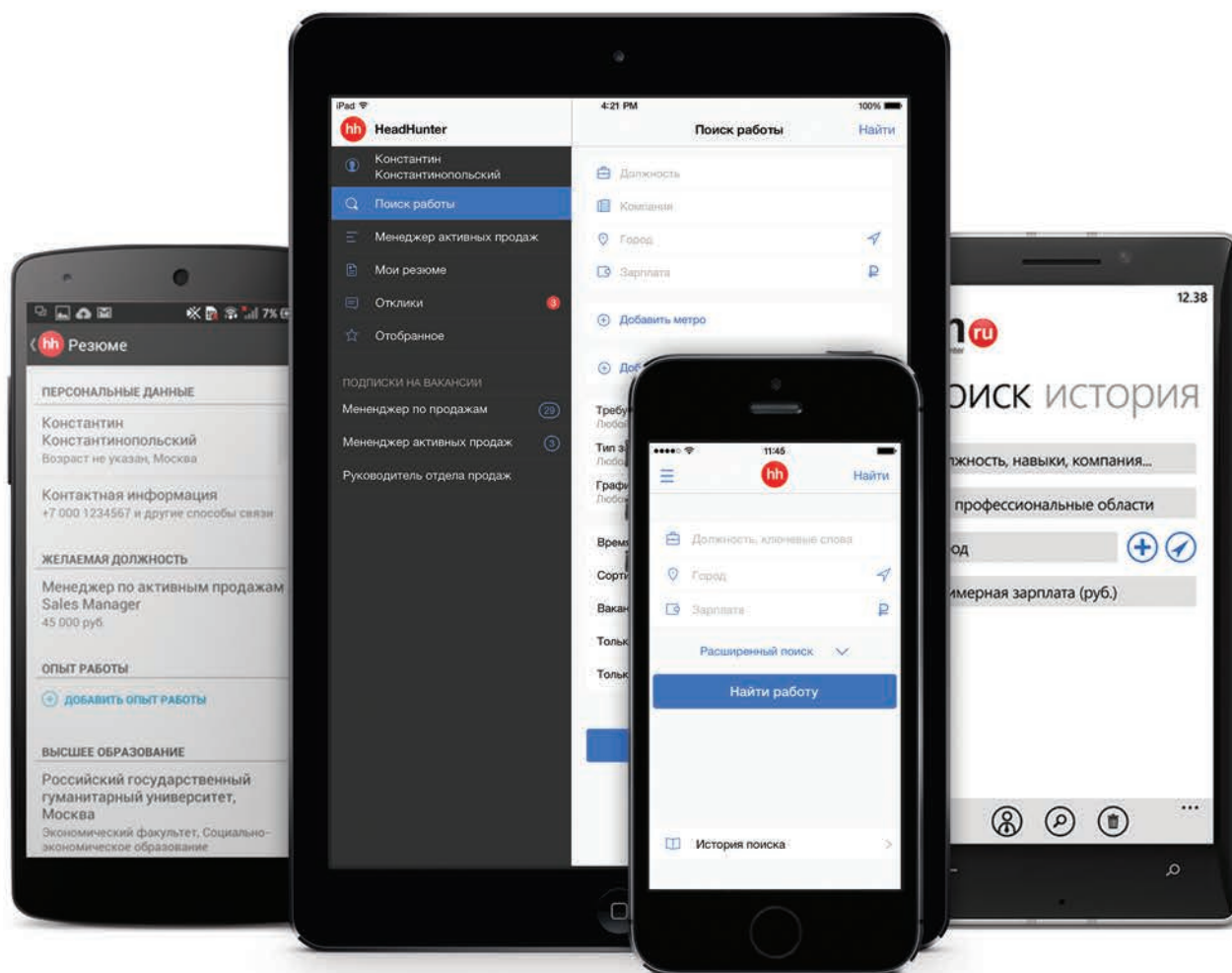
➤ В ядре Linux 3.19 46% изменений затронули драйверы устройств, 19% — аппаратные архитектуры, 12% — сетевой стек и по 4% — файловые системы и подсистемы ядра. Источник: kernel.iu.edu

➤ В X.Org Server 1.17 интегрирован DDX-драйвер xf86-video-mode-setting и оптимизирована архитектура 2D-ускорения Glamor. Источник: lists.x.org

0+



Работа мечты в твоём кармане!

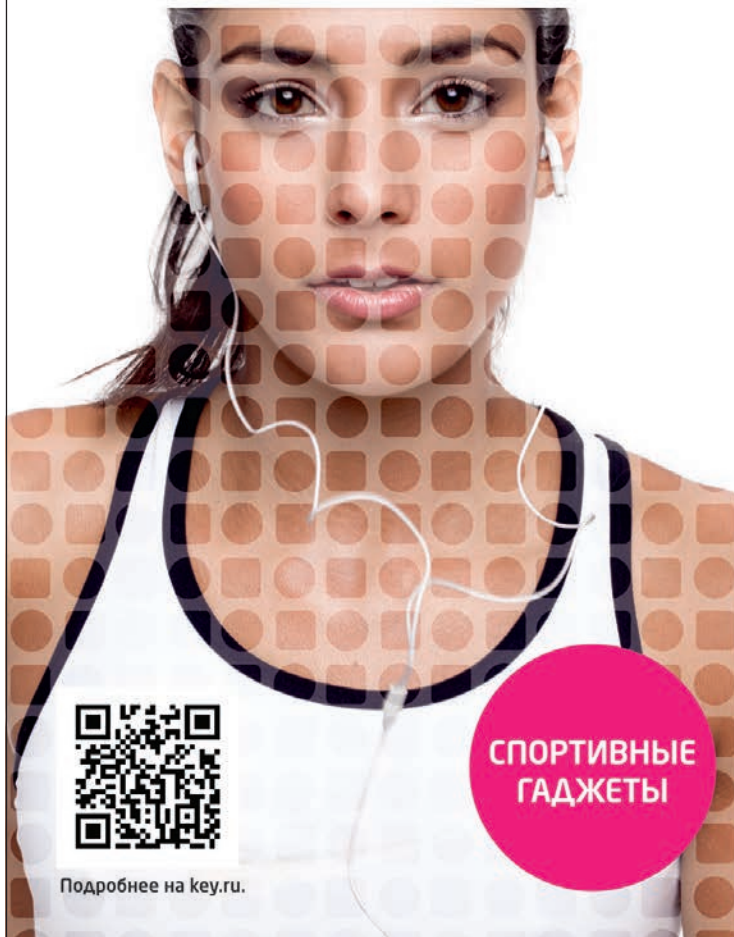


Искать просто, откликаться легко.

<http://hh.ru/mobile>



ТВОИ ПОМОЩНИКИ В ТРЕНИРОВКАХ



Экшн - камера



Экшн-камера Sony HDR-AS30

Smart - часы



Smart-часы Cookoo Watch

Smart - здоровье



Браслет Jawbone UP



Шаромер Misfit Shine

СПОРТИВНЫЕ
ГАДЖЕТЫ



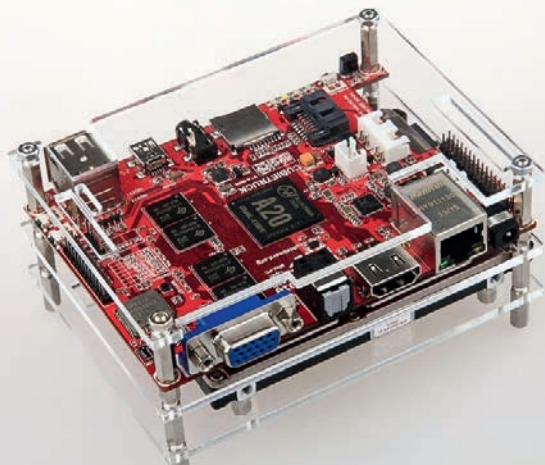
Подробнее на key.ru.

Товар сертифицирован. Количество товара ограничено.



Одноплатный компьютер Cubieboard3

Cubietruck



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- Процессор: AllWinnerTech SOC A20, ARM Cortex-A7 Dual-Core
- Видео: ARM® Mali 400 MP2
- Выходы: HDMI, VGA 1080p
- Оперативная память: 2 Гб, DDR3, 480 МГц
- Встроенная память: 8 Гб NAND
- Внешние накопители: интерфейс SATA 2.0 для 2,5" HDD; возможно подключение 3,5" HDD с использованием внешнего питания 12 В
- Сеть: 10M/100M/1G Ethernet, Wi-Fi и Bluetooth
- Питание: DC 5 В, 2,5 А с подключенным HDD; возможно подключение литиевого элемента питания
- Периферия: 2 × USB host, 1 × USB OTG, 1 × Toslink (SPDIF), 1 × IR, 4 × LEDs, 3,5" аудиовыход, 3 кнопки
- Поддерживаемые ОС: Android, Linux-based, BSD
- 54 контакта расширения I2S, I2C, SPI, CVBS, LRADC × 2, UART, PS2, PWM × 2, TS/CSI, IRDA, LINEIN & FMIN & MICIN, TVIN × 4
- Размер платы: 11 × 8 см

КОМПЛЕКТАЦИЯ

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

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

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



LINUX FORMAT

Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Matuntu — гуманизм по-нашему

Как известно, самый гуманистический дистрибутив всех времён и народов, под именем Ubuntu, возник в Южной Африке, сбросившей цепи апартеида. Однако и наша страна, с её давними гуманистическими традициями, не осталась в стороне. И в ней на базе Ubuntu были созданы сначала Runtu, а за ней — Matuntu. Последняя базируется на текущих LTS-релизах «головного» дистрибутива, отличаясь такими особенностями: применением Mate, наследника Gnome 2, в качестве рабочей среды; включением её наработок, уже не поддерживаемых в апстриме, но сохраняющих интерес; и «отвязкой» от метапакетов Mate, что позволяет каждому применителю индивидуализировать систему по своему вкусу. И, конечно, безупречная локализация и адаптация к российским реалиям для дистрибутива подразумевается по умолчанию. Подробнее о Matuntu см. на стр. 60.

Matuntu распространяется в виде гибридных ISO-образов, из которых актуальны четыре, на базе Ubuntu 14.04.1: 32-битная и 64-битная пары. Сборки для одной архитектуры различаются версиями Mate (1.6 и 1.8) и включением/отсутствием Compiz по умолчанию. Все они штатно укомплектованы массой приложений — не обязательно обычных, но всегда интересных. А сама среда Mate здесь точно воспроизводит «второгном» на пике его развития, причём в очень аккуратном исполнении. Добыть ISO-образы можно на сайте проекта: matuntu.ru. И там же на форуме получить дружескую помощь в освоении дистрибутива. alv@posix.ru

Сегодня мы рассматриваем:

- AV Linux 6.0.4** 14
AV значит «аудио-видео» — отсюда ясно, что это дистрибутив для творческих личностей с музыкальным уклоном. И действительно, он содержит синтезаторы, секвенсоры, редакторы... притом в ассортименте.
- The SSS 10.1** 15
А вот S значит «сервер», и название дистрибутива The Smallest Server Suite — явно в пику Microsoft. Развертывание сервера здесь секундное дело, а весит он чуть больше 50 МБ.
- Toshiba Chromebook 2** 16
Наконец-то экран хромбука стал достойным видео качества 1080p! Правда, браузерные игры придется
- выбирать попроще, не то кадры начнут спотыкаться.
- Intel Core i7-5820K** 17
Для тех, кто не тратит заработанные непосильным трудом денежки зря: этот процессор по производительности почти не отстает от старших братьев, но намного их дешевле.
- Samsung 845DC** 18
Твердотельный диск класса предприятия способен писать-читать до 165 Гб ежедневно целых 5 лет (срок действия гарантии). Это ли не пример должительства?
- BitScope BS05** 19
Миниатюрный — меньше не бывает — осциллограф удивит вас своим
- обширным набором функций. Ах, какая штука...
- Lumino от 4tronix** 20
Еще один шустрый робот на колесах, но не на Raspberry Pi, а на Arduino. Сообщество Arduino наработало массу интересных проектов — так не пропадать же добру.
- Geometry Wars 3** 21
С этой 3D-стрелялкой придется выучиться не мигать... иначе ваш конец близок: действие очень бурное.
- Броневые SSL и TLS** 21
Книга для системных администраторов, среди прочего, описывает реальные атаки, известные по документам Сноудена.



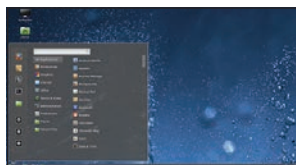
› Цветовая гамма, видимо, обязана навевать вдохновение к скрипичным концертам.



› Можно не только любоваться красками заката, но и смотреть видео высшего качества.

Сравнение: Рабочие столы с. 24

Cinnamon



Enlightenment



Gnome Shell



KDE Plasma



Mate



Выбор рабочего стола очень важен: ведь именно через него мы общаемся со своим ПК. Еще недавно пальму первенства по очереди вырывали друг у друга Gnome и KDE. Но им приходится потесниться!

AV Linux 6.0.4

На сей раз автор миллиона тайных фантазий **Шашанк Шарма** хочет раскрыть свои таланты в создании видео, графики и звука с помощью AV Linux.

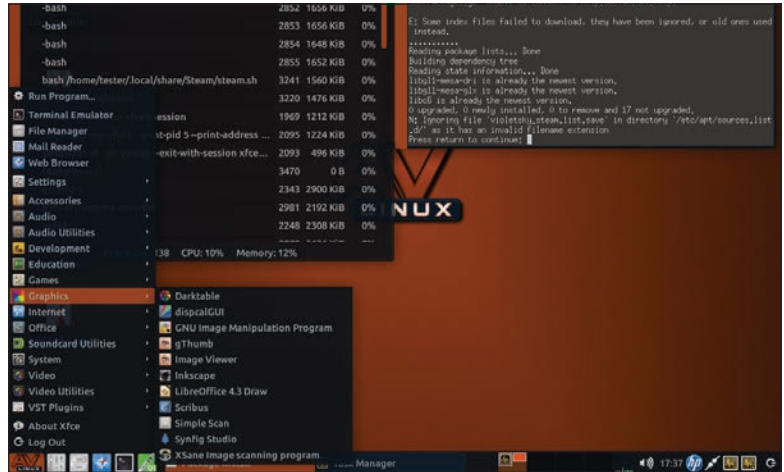
Кратце

» Редактируете ли вы файлы или создаёте их с нуля, AV Linux предлагает большую подборку инструментов для аудио, видео и графики в едином крепко сбитом дистрибутиве. См. также: [Ubuntu Studio](#).

Для большинства пользователей Linux выбор подходящего дистрибутива — дело весьма непростое. Но это не всегда так. AV Linux — дистрибутив на базе Debian, ставящий своей целью упростить творческую часть Linux-сообщества. Следуя философии «Установи и твори», он вмещает громадную коллекцию инструментов для создания аудио-, видео- и графического контента. Первые выпуски дистрибутива базировались на Ubuntu Studio, но сейчас он основан на Xfce-версии Debian. Дистрибутив, по его же признанию, разработан с целью превратить старые ПК в рабочие станции мультимедиа. Хотя он хорошо приживётся и на новых машинах, но всё же выпущен только в 32-битной версии с поддержкой PAE. И если вы хотите запустить его в *VirtualBox*, перед загрузкой позаботьтесь включить опцию PAE, а это ограничивает каждое приложение 4 Гб памяти.

Устанавливаемый live-дистрибутив может быть запущен с DVD или с USB-флэшки; на сайте есть также инструкции по запуску с USB с сохранением данных [persistence mode]. Руководство в формате PDF также кратко описывает установку с помощью *Remastersys* и знакомит с инструментами. Трудностей не будет: мастер задаёт ряд вопросов, проводя вас через процесс, а для разбиения диска на разделы в дистрибутиве есть *GParted*.

Два года назад релиз AV Linux 6.0 был объявлен последним, но проект вновь расцвёл: нынешний релиз — уже третий в 2014-м году.



» Большинство приложений открыты, но есть и проприетарные демо-версии.

Если у вас вдруг возникнут проблемы с загрузкой обычного live-окружения, дистрибутив предлагает загрузку в режиме совместимости. От режимов восстановления в большинстве дистрибутивов он отличается тем, что пропускает некоторые опции загрузки, способные конфликтовать с определённым оборудованием, поэтому его можно использовать при неполадках.

Инструменты ремесла

Хотя основное внимание и уделено программам для творчества, дистрибутив подходит для ежедневных задач. Он включает все привычные приложения, такие как *Audacity*, *Cheese*, *Chrome*, *Firefox*, *LibreOffice*, *Transmission*, *VLC Player*, *Xchat*, *Xine* и другие. Из меню Игры можно даже установить *Steam*. Так как дистрибутив основан на Debian, предусмотренный менеджер пакетов *Synaptic* позволит без труда устанавливать дополнительные приложения.

У дистрибутива собственное меню, где приложения аккуратно разложены по категориям: Интернет, Аудио и т. д. Меню также включает ряд консольных приложений, таких как *JackCapture*, *Patchage*, *WineASIO* и других в разделе Аудио Утилиты, где вы можете запустить их напрямую, не открывая предварительно терминал.

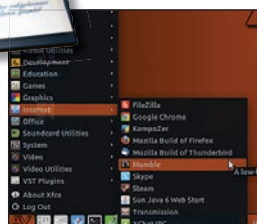
Все приложения, вошедшие в дистрибутив, и не перечислять; например, он может похвастаться коллекцией синтезаторов, секвенсоров, редакторов и пр. Также там имеются микшер аудио/видеопотоков *FFMux*, инструмент *DeVeDe* для записи DVD, видеоредактор *Cinelerra*, DVD-риппер

и перекодировщик *Handbrake*, драм-машина *Hydrogen*, мультимедийный пакет *LMMS*, пакет для моделирования и анимации *Blender3D*, *Darktable* для фотозентузиастов, *Inkscape*, *Scribus* и ещё много замечательных приложений. Их описания появляются по наведению мыши на пункт меню. Кроме того, дистрибутив предлагает демо-версии ряда проприетарных инструментов и плагинов, например, *Pianoteq* для моделирования пианино и *Renoise* для синхронизации аудио.

Благодаря наследию Debian, а также подробной документации, форуму и отзывчивому сообществу, использовать AV Linux очень легко. Этого, пожалуй, не скажешь обо всех входящих в него инструментах — но исключительно по причине их богатых возможностей. **LXF**



Свойства новскидку



Уже настроено

Приложения заранее настроены и готовы к работе. Настройки по умолчанию подойдут для большинства пользователей-творцов.



Платные инструменты

Включает демонстрации коммерческих несвободных программ. Информация о лицензировании — в PDF-файле EULI на сайте.

LINUX FORMAT Вердикт

AV Linux 6.0.4

Разработчик: Глен Мак-Артур

[Glen MacArthur]

Сайт: <http://bit.ly/AVLinuxOS>

Лицензия: GPL и проприетарные

Функциональность 8/10

Производительность 6/10

Удобство в работе 8/10

Документация 8/10

» Первоклассный дистрибутив для творцов, хотя его приверженность 32 битам налагает ограничения на скорость и функции.

Рейтинг 7/10

The SSS 10.1

Шашанк Шарма обнаружил, что с дистрибутивом The Smallest Server Suite можно разворачивать серверы за секунды, и очень этим впечатлён.

Вкратце

» Дистрибутив, позволяющий запустить различные типы серверов за считанные секунды. Включает антивирус, менеджер разделов, средства мониторинга сети и другие необходимые инструменты. См. также: 4MLinux.

The Smallest Server Suite [англ. Маленький Серверный Комплект] — легковесный дистрибутив на базе 4MLinux. Последний фокусируется на предоставлении четырёх основных функций — Восстановлении, Мульти-медиа, Мини-сервере и, гм, Мистериях (просто играх, на самом деле), а The SSS идеален для разворачивания серверов и их обслуживания.

Дистрибутив доступен в трёх редакциях как устанавливаемые live-образы. «Базовый» релиз (примерно на 30 МБ) — самый лёгкий и может применяться для серверов FTP, HTTP, SSH и Telnet. PHP-вариант (52 МБ) включает всё вышеперечисленное плюс PHP, MariaDB и Apache.

Третья редакция, которую мы и рассматриваем в этом обзоре — Toolbox. Она включает всё из первых двух редакций, а также инструменты для резервного копирования, мониторинга системы и сети, менеджер разделов *Ranish Partition Manager*, антивирус *ClamAV* и прочее. Все три версии также имеют прокси-сервер *Polipo* и анонимайзер *Tor*. Toolbox доступен как устанавливаемый live-образ размером 95 МБ, который можно записать на диск или флэшку (через *UNetbootin*).

Сила в лёгкости

Последний выпуск The SSS основывается на 4MLinux 10.1 Server Edition. Дистрибутив не предлагает GUI и крайне экономичен в выборе инструментов. Например, в него не включены man-страницы установленных инструментов.

```
List of servers:
ftpd --> vsftpd (via inetd)
httpd --> Apache (standalone)
proxy --> Polipo (standalone)
sshd --> OpenSSH (via inetd)
telnetd --> BusyBox (via inetd)
firewall --> iptables (4MLinux Firewall)

An example how to use them:
ftpd start --> start FTP server
ftpd stop --> stop FTP server
ftpd restart --> restart FTP server
ftpd test --> test FTP server

Another example:
serverd start --> start all servers
serverd stop --> stop all servers
serverd restart --> restart all servers
serverd test --> test all servers

PHP support in Apache (if installed) is enabled by default.
Stunnel and MariaDB (if installed) are started automatically by httpd.
See also: /etc/firewall /etc/httpd /etc/inetd /etc/php /etc/polipo /etc/server
root@TheSSS:~#
```

» SSS — надёжное средство настройки серверов, создания резервных копий, проверок на вирусы, мониторинга сети и много чего ещё, и всё это на одной удобной флэшке.

Когда вы загружаете образ, загрузчик предлагает пять вариантов. Первый — The SSS 10.1 PHP, идентичный PHP-редакции. Вторым в списке идёт *BakandImagCD 10.1*, предлагающий инструменты резервного копирования данных и работы с образами дисков. Этот вариант можно выбрать, чтобы скоренько сделать бэкап. Третий вариант — *AntivirusLiveCD*. Выберите его, если хотите запустить *ClamAV*. Приложение уже установлено и настроено, но при входе необходимо будет обновить сигнатуры. Чтобы получить инструкции, введите **help** в командной строке. Оставшиеся два пункта меню — *TestDisk* и *Ranish Partition Manager*.

Загрузившись в The SSS, первым делом создайте пароль для root. Войдя, наберите **help**, и увидите три пункта дальнейших действий: *server*, *netconfig*, *install2hd*. Для получения списка доступных серверов и их запуска/остановки наберите **server**. По умолчанию никакие серверы не запущены, но это исправимо: отредактируйте файл */etc/server/autostart.conf*. Если соединение с сетью не сконфигурировалось само, наберите **netconfig**. Скрипт задаст вам вопросы об интерфейсе, DHCP и т.д. и настроит устройство. Инструкции элементарны, но для почина достаточны.

Чтобы установить дистрибутив на диск, запустите скрипт *install2hd*. Он задаст обычные вопросы о диске и разделе для установки. Установщик не умеет переразбивать диск, но для обустройства пространства The SSS включает *Fdisk* и *Cfdisk*.

Также дистрибутив включает несколько инструментов мониторинга: *nmon*,

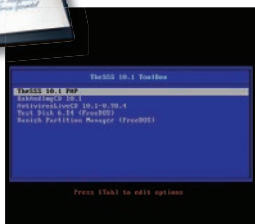
nmonitor, *iptraf*, *wavemon* и другие. Для удобства редактирования файлов настроек дистрибутив содержит менеджер файлов *Midnight Commander*; для web-серфинга послужит браузер *Links*.

Дистрибутив не сильно поменялся за годы — это видно по официальному руководству, каковым служит пост в блоге от 2013 года. Единственный источник документации — официальный блог, где много постов о всяких функциях, в том числе об FTP-, SSH-, прокси-серверах и так далее. В некоторых из них говорится о 4MLinux, но они применимы и к The SSS, так как управление двумя дистрибутивами очень похоже.

Дистрибутив идеально назван, и мы рекомендуем The SSS за его простоту. Это хорошо продуманный и очень способный дистрибутив. **LXF**



Свойства навскидку



Опции загрузки

Предлагает специализированные образы для резервного копирования на USB или на FTP-сервера, а также для запуска *ClamAV*.



Много инструментов

Невероятный ассортимент инструментов и утилит. Не помешал бы полный список включённых программ.

LINUX Вердикт
FORMAT

The SSS 10.1 Toolbox

Разработчик: Збигнев Конояцки
[Zbigniew Konojacki]
Сайт: <http://thesss.4mlinux.com>
Лицензия: GPLv3

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

» «Самомалейший Серверный Комплект» оправдывает своё название; этот легковесный дистрибутив обязан быть под рукой у каждого админа.

Рейтинг 9/10

Toshiba Chromebook 2

Наконец-то у хромбуков появился экран для 1080p, а также элегантный дизайн, заслуживший похвалу геймеров в лице **Кевина Ли**.

Вкратце

» Улучшенный Chromebook от Toshiba, с более тонким дизайном, приемлемой жизнью аккумулятора и 1080p-дисплеем.

Спецификация

- » Процессор: Intel Celeron N2840, 2,16 ГГц
- » Графика: Intel HD Graphics
- » Оперативная память: 4 ГБ
- » Экран: 13,3 дюймов, 1920×1090
- » Хранилище: 16 ГБ SSD
- » Разъёмы: 1×USB 3.0, 1×USB 2.0, HDMI, SD card
- » Связь: Wireless-AC, Bluetooth 4.0, web-камера HD
- » Габариты: 320×213×19,3 мм, 1,34 кг

На бумаге Toshiba Chromebook 2 выглядит улучшенным по всем параметрам по сравнению с оригиналом, но не принёс ли он что-то в жертву ради яркого экрана? Toshiba Chromebook 2 одновременно меньше и тоньше своего предшественника, несмотря на такой же 13,3-дюймовый экран. Благодаря новому дизайну без кулера Toshiba смогла уменьшить габариты с 20,3 мм до 19,3 мм; заодно уменьшился и вес хромбука — с 1,5 кг до 1,34 кг.

Если отвлечься от цифр, Toshiba Chromebook 2 — это гладкий корпус, переходящий в заметно более тонкую грань. Toshiba немного улучшила дизайн Chromebook 2 матовой окраской и спрямлёнными углами, которые придают Chromebook 2 более чёткий вид. Единственный заметный огрех в дизайне — белый пластиковый низ, не сочетающийся с серебристым металлом остального корпуса. Клавиатура более чем удобна и имеет традиционную раскладку, а под ней находится большой трекпад для точных щелчков.

Toshiba снабдила свою новую модель процессором Intel Bay Trail, 1080p-дисплеем и 4 ГБ оперативной памяти, но всё это повлекло и более высокую цену. Стоимость в США — \$329, или около £249.

После раздражающе-неторопливой работы Samsung Chromebook 2 мы боялись, что Toshiba Chromebook 2 не покажет хорошей производительности. У обоих ноутбуков один и тот же процессор Intel Celeron N2840, и при этом Toshiba решила поставить дисплей с Full HD. К счастью, такие



» Новый Toshiba Chromebook изящен благодаря отсутствию вентилятора.

проблемы практически отсутствовали. Ноутбук гладко работал с 18 вкладками браузера под музыку, играющую в фоне, и справился с 1080p-видео без заиканий или задержек потока.

Серьёзная мощь

Мы заметили проблемы с производительностью, только когда решили поиграть в браузерную игру *Bastion* на полном разрешении. По мере того, как мы продвигались вперёд, а игровой мир валился с неба, частота кадров упала до прихрамывающих 10–15 кадров в секунду. Это не так уж удивительно, если учесть, что игра требует как минимум 512 МБ видеопамати, хоть она и браузерная. Что ж, найдутся браузерные игры и попроще.

Если хромбукам раньше чего-нибудь и недоставало (если не говорить о приложениях), то это разрешение экрана. Хромбуки долго обходились жалким разрешением 1366×768. Toshiba Chromebook 2 предоставляет не только больше пикселей, но и в целом лучшее качество экрана. Вместо использования TN-панели, как у большинства хромбуков, Toshiba выбрала дисплей IPS, который отображает живые цвета и глубокий чёрный. В результате — это первый испробованный нами хромбук с великолепным дисплеем,

на котором действительно хочется смотреть кино.

Конечно, обратная сторона такого изобилия пикселей — время жизни аккумулятора. Toshiba 2 продержался всего 6 часов 26 минут браузинга, включая небольшую видеоконференцию и просмотр фильмов. Само по себе это неплохо, но мы ожидали от хромбука большего долгожительства, как, например, у Acer C720 (8 часов 30 минут) или Samsung Chromebook 2, который держится на час меньше (7 часов 32 минуты). **LXF**



Свойства навскидку



Экран IPS

Toshiba присоединяется к пока еще немногочисленному клубу хромбуков, обладающих ярким 1080p-дисплеем.



Трекпад и клавиатура

Прочная клавиатура вместе с большим и точным трекпадом делают Toshiba хорошим для почти любой продуктивной работы.

LINUX FORMAT Вердикт

Toshiba Chromebook 2

Разработчик: Toshiba
Сайт: www.toshiba.co.uk/chromebook
Цена: £249

Функциональность	6/10
Производительность	8/10
Удобство в работе	9/10
Оправданность цены	8/10

» Хотя и коротка его жизнь от аккумулятора, зато этот лёгкий 13-дюймовый хромбук имеет самый приятный экран среди всех.

Рейтинг 8/10

Intel Core i7-5820K

Дэйв Джеймс, кажется, отыскал недостающее звено между Haswell и Haswell-E и расскажет, почему новый продукт Intel придаст мощности вашему Linux.

Спецификация

- » Сокет: Intel LGA 2011-v3
- » Технология ядра: Intel 4th Gen Core
- » Тактовая частота: 3,3 ГГц
- » Турбо: 3,6 ГГц
- » Ядер: 6
- » Поток: 12
- » Печатный процесс: 22 нм
- » Кэш: 15 МБ
- » TDP: 140 Вт
- » Каналов памяти: 4

В передовых чипах платформы Intel Haswell-E сосредоточены все новейшие технологии текущего поколения вычислений. Если вы получаете быстрое PCIe-соединение с памятью, новое поколение системной памяти и восьмиядерный процессор, объединяющий всё это, у вас будет апофеоз сегодняшних технологий для настольных ПК.

Вполне справедливо, что все заголовки прессы отобрал Core i7-5960X со своими 16 потоками вычислительной мощи. Но для нас шестиядерный Core i7-5820K — самый интересный процессор в новом поколении чипов, учитывая его выгодную цену.

Мы в Linux Format довольно-таки прижимисты — на нашу скудную зарплату купишь разве что миску похлёбки, продаваемой в подвальной столовке Башни издательства Future со скидкой для сотрудников — поэтому проспектик шестиядерного чипа Intel выглядел соблазнительно. В прошлый раз эту сомнительную похвалу заслужил Core i7-4930K из прошлого поколения, известный как Ivy Bridge-E второго класса, хотя на старте он стоил £500.

Core i4-5870K более чем на £200 дешевле прежнего чипа — и при этом заметно шустрее. Фактически он даже превосходит Core i7-4960X — топовый процессор Ivy Bridge-E — хотя тот стоит £800. 5820K вобрал в себя всё от Haswell-E, используя такую же схему, что и крутой 5960X. То есть у него та же 22-нанометровая архитектура 4-го поколения

ядер, и он подходит для той же платформы X99, что и его старшие братья.

Всё, что вы утрачиваете по отношению к i7-5960X, это те самые два ядра, 5 МБ кэша и 12 линий PCIe. По сравнению с i7-5930K за £450 (шестиядерным средним из трёх братьев Haswell-E), вы просто получаете слегка более медленный чип и чуть меньше линий PCIe.

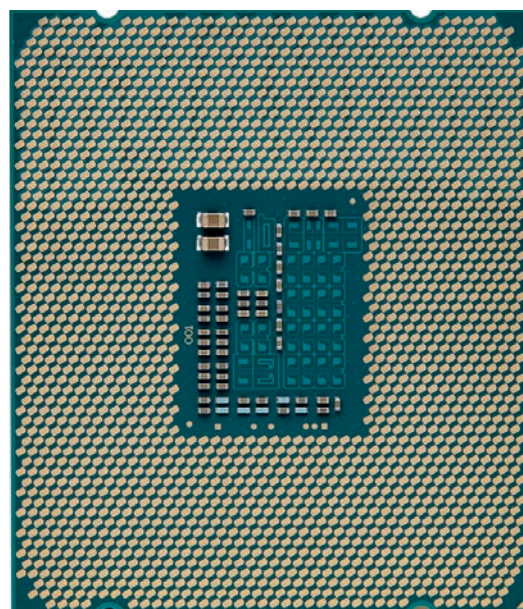
Получается, этот процессор вполне серьёзен. Мы говорим о приличном чипе Intel, а стоит он всего лишь чуть больше, чем четырёхядерный Devil's Canyon из последнего стандартного обновления Haswell. Но доступные шестиядерные CPU — не новость, верно? В конце концов, вы можете прямо сейчас взять шестиядерный AMD за £75.

Чипы не бесплатны

Штука в том, что AMD и Intel придерживаются разных представлений о способах подсчёта ядер; иначе мы бы назвали 5820K 12-ядерным. Модульная архитектура CPU, которую AMD реализовала в Bulldozer, был ответом AMD на HyperThreading от Intel, но с чуть более актуальным чипом, добавленным в многоядерную смесь. Однако AMD, похоже, сочла тот путь ошибочным и вернулась к более традиционным CPU, когда несколько лет назад свет наконец увидела архитектура Zen.

Не жертвуя реальной производительностью, вы можете построить шестиядерную X99-систему всего на £100 дороже, чем четырёхядерная машина на Canyon Z97 от Devil. Хотя цены на DDR4 высоки, но если вы хотите установить 16 ГБ ОЗУ — а вы, скорее всего, хотите этого, если собираетесь использовать всю силу 12 вычислительных потоков — то нет большой разницы между DDR3 на 16 ГБ с 1600 МГц и DDR4 на 16 ГБ с 2133 МГц.

5820K по-прежнему располагает всего 28 линиями PCIe 3.0 и не способен работать с двумя графическими картами на скорости x16: подобное могут



» Серьёзная мощность, причём за приемлемые деньги.

себе позволить только 5930K и 5960K с 40 линиями. Но с учётом того факта, что у PCIe 3.0 пропускная способность больше, чем используют наши карты, разница между x8 и x16 для них почти отсутствует. Отсутствие поддержки SLI/CrossFire в большинстве игр для Linux тоже уменьшает масштаб проблемы. Мощный шестиядерный CPU в ближайшее время не собирается уступать кому-нибудь своё место — коль скоро Broadwell-E отложен до 2016 года — и с этой платформой вы получите серьёзную вычислительную мощь на несколько лет. **LXF**



Свойства навскидку



Скорость в играх

Очевидно, что король вычислений — восьмиядерный 5960X, но гораздо более дешёвый 5820K не намного слабее.



Дьявольски быстро

Хорошо разгоняется, легко достигая 4,45 ГГц — где уж там 5960X с его мизерными 4,375 ГГц, да ещё за такие-то деньжищи!

LINUX FORMAT Вердикт

Intel Core i7-5820K

Разработчик: Intel
Сайт: www.intel.com
Цена: £294

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

» Доступный и супермощный, он выдержит проверку временем в сравнении с четырёхядерными процессорами.

Рейтинг 9/10

Samsung 845DC

845DC EVO — твердотельный диск класса предприятия с соответствующим ценником. **Орестис Бастунис** хочет узнать, находится ли он на Звёздном Пути.

Вкратце

» Предлагаемая новую TLC-флэш-память, Samsung надеется, что получил SSD с большой ёмкостью и готовый к высокоскоростному чтению и длительному использованию на Linux-серверах с интенсивной нагрузкой. См. также: Samsung 850 Pro и Crucial MX100.

Samsung 845DC (Data Centre — «центр данных») EVO — это твердотельный диск (SSD) класса предприятия, оптимизированный под большую нагрузку на чтение данных, как это часто бывает в базах данных и серверных приложениях. Здесь, как и в 840 EVO и PM843, используется флэш-память TLC или, как её предпочитает называть Samsung, 3-битная MLC NAND. TLC-память дешевле в производстве, чем 2-битные MLC NAND, используемые в большинстве накопителей для пользователей, но имеет и минусы, а именно — меньшие выносливость и производительность.

Модель на 480 ГБ, которую мы тестировали, стоит примерно £340, а её ёмкость составлена из четырёх чипов по 128 ГБ, при этом 32 ГБ составляет резервная область. Объём LPDDR2-кэша — 512 МБ, а у модели на 960 ГБ он составляет 1 ГБ. Как и раньше, Samsung использует Toggle 3-bit MLC NAND с трёхядерным контроллером MEX. Подобно Samsung 850 Pro, контроллер разгоняется до 400 МГц. Гарантия действует пять лет — далековато от тех десяти лет, которые Samsung даёт для 850 Pro.

Важное дополнение к этой модели — защита от сбоя питания, обеспеченная дополнительным набором конденсаторов внутри накопителя. Если 845DC EVO по какой-либо причине отключается от питания, эта особенность даст ему достаточно времени, чтобы сбросить данные из кэша в хранилище.

Любая флэш-память может быть записана лишь ограниченное количество раз,

которое определяется как циклы программного стирания (program-erase, P/E). Это неизбежно. Чтобы стереть ячейку, на неё необходимо подать напряжение. При этом каждый раз ячейка становится чуть менее чувствительной к диапазону напряжений, которые она способна принимать (и которые представляют двоичные значения). В конце концов ячейка становится нерабочей.

Естественно, для накопителя SLC подобного объёма необходимо потратить немало кремния флэш-памяти, что взвинчивает цену. Но потребности накопителей TLC гораздо меньше. В SCL ячейка выдерживает 1 000 000 циклов P/E, а для среднего чипа TLC эта цифра падает до 1000. В 845DC EVO Samsung изловчился утроить её, доведя до примерно 3000 циклов P/E на каждую ячейку.

Так как Samsung 845DC EVO — накопитель промышленный, он был разработан с прицелом на максимальный срок службы, несмотря на то, что это TLC NAND. Надёжность SSD может быть измерена множеством способов, но стандартная характеристика, которую сообщают большинство изготовителей — это TBW, или общее число записанных байт. Для большинства SSD оно удваивается вместе с ростом ёмкости. Модель 845DC EVO на 240 ГБ оценивается в приличные 150 ТБ записей, на 480 ГБ — в огромные 300 ТБ, а самая большая модель, на 960 ГБ — в действительно колоссальные 600 ТБ.

Долгожитель

Это всё очень полезно. Если взять в качестве примера 480-ГБ модель, то вы получаете 60 ТБ записи на каждый из пяти лет, в течение которых действует гарантия, то есть 164,38 ГБ каждый день без исключения. Если установить 845DC EVO как системный раздел в ваш домашний ПК, то простым «повседневным» использованием достичь этого лимита будет сложно.

Samsung прислал нам пару 480-ГБ 845DC EVO на пробу, и мы протестировали их и в отдельных конфигурациях, и в RAID 0. Производительность заявлена



» Для большинства домашних пользователей — долгожитель, но есть SSD и побыстрее.

как 530 МБ/с для чтения и 410 МБ/с для записи. Количество случайных чтений 4K QD32 в секунду заявлено как 87 000, а случайных записей — много меньше 14 000. Результаты наших тестов последовательного чтения совпали с данными Samsung, достигнув 543 МБ/с. Скорость записи оказалась 438 МБ/с. В RAID 0 эти скорости практически удвоились, как и ожидалось: 1117 МБ/с на чтение и 873 МБ/с на запись.

Результаты позволяют причислить 845DC EVO к верхнему сегменту по чтению, но лишь к середине чартов SSD по записи. Другие марки устройств превзойдут его по обоим фронтам, и всё же этот накопитель очень быстрый. **LXF**

Свойства навскидку



SSD-долгожитель

По методологии тестирования JEDEC, минимальная жизнь этого Samsung составил 22 года.



Новый NAND

Чипы NAND бережно отобраны среди самых качественных пластин NAND с 3 битами на ячейку.

LINUX FORMAT Вердикт

Samsung 845DC EVO

Разработчик: Samsung
Сайт: www.samsung.com
Цена: £340

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

» Обеспечивает очень быстрое чтение и экстраординарную долговечность, но уж слишком дорог.

Рейтинг 8/10

BitScope BS05

Джонни Бидвелл попробовал и увидел, как много можно сделать с самым маленьким из известных вам осциллографов.

Вкратце

» Миниатюрный, но полностью функциональный USB-осциллограф для смешанных сигналов и анализатор логики и спектра, специально разработанный для проектов на Raspberry Pi.

ВitScope — авторитетная австралийская компания, более 15 лет выпускающая осциллографы. Их последний продукт, BitScope Micro, изготовлен специально для Raspberry Pi: он маленький, работает от USB и напичкан возможностями. Он также не боится воды благодаря своей пластиковой обёртке. Хотя он и не похож на громоздкие двухлучевые осциллографы, которые любил ваш школьный учитель физики (простите великодушно, мистер Уоллес), он способен делать всё то же, что делают они, и даже больше. И весит всего 12 г.

Помимо возможности уловить два аналоговых канала (и таким образом создать знаменитые фигуры Лиссажу), он способен произвести их частотный анализ. Более того, он имеет шесть специальных логических каналов для декодирования последовательного протокола, SPI, I2C и CAN. Вы можете получить ещё два дополнительных канала с помощью компараторов аналогового сигнала. Он также работает как генератор сигналов и импульсов: входящее в комплект программное обеспечение позволяет генерировать синусоидальные, прямоугольные и пилообразные волны с частотами от 4 до 16 кГц и амплитудами до 3,3 В. А если у вас есть небольшие навыки программирования, то устройство покажет вам произвольные волны, определённые 1024 точками. Подсоединив (одним из 10 предоставленных кабелей для захвата) пин L5 (для импульсов) или пин L4 (для волн) к входным каналам, вы даже сможете отображать сигнал по мере его генерации.



» Это не осциллограф, вот это — осциллограф, как сказал бы Пол Хоган [Paul Hogan — австралийский актер, играл в фильмах о «Крокодиле Данди»].

Хотя программы, предоставляемые с BitScope, выглядят непритязательно, они достаточно мощные, и компания определённо приложила усилия, чтобы они работали эффективно, в том числе на Pi. Основное приложение, *BitScope DSO*, имеет все элементы управления, которые нашлись бы в обыкновенном настольном осциллографе, а также те, которые там не нашлись бы: например, опции для сглаживания или затухания отображаемых волн. Также вы можете загрузить приложения *Chart* (для записи данных), *Logic* (для временного анализа протоколов и логики) и *Meter* (для автоматизированных измерений или для использования датчиков в качестве вольтметра). Программы от BitScope, благодаря толковому подходу к проектированию, работают со всеми продуктами компании, и пакеты доступны для Mac, Windows, Raspberry Pi и Ubuntu. Кроме того, доступен обычный исполняемый файл для Linux и исходные коды всего пакета.

Осциллограф имеет впечатляющую 50-герцовую частоту кадров, которая рендерится в реальном времени.

Виртуально всемогущ

Все продукты BitScope построены на базе виртуальной машины BitScope, которая использует скрипты для управления регистрами. Это означает, что энтузиасты могут написать код для нужд конкретного проекта и запустить его прямо на устройстве. Также имеется хорошо задокументированный API BitLib, позволяющий юным инженерам писать код на C, C++, Python или Pascal. Приложения и API

поддерживают удалённые измерения, делая осциллограф ещё более разносторонним средством — неважно, используете ли вы его как основную часть своего проекта или только для диагностики проблем внутри.

BS05 доступен в США с прошлого апреля, но при цене в \$150 (плюс таможенная пошлина для жителей Великобритании) он оказался дороговат. Теперь он официально появился в Великобритании, и в Farnell (element14) его можно приобрести за £89.

Часть потенциальных пользователей, возможно, удержит от покупки пока ещё увесистый ценник, а матерые электронщики уже, скорее всего, и так имеют подходящий им инструментарий для анализа сигнала. Но если вы ещё только приступаете к делу, этот приборчик будет отличной инвестицией. **LXF**



Свойства навскидку



ПО DSO

Пусть оно неказисто с виду, но умеет делать намного больше, чем обычный осциллограф.

Анализ логики

Исследуйте смешанные сигналы, декодируйте протоколы, записывайте входящие данные, причём всё сразу.

LINUX FORMAT Вердикт

BitScope BS05

Разработчик: BitScope
Сайт: www.bitscope.com
Цена: £89

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

» Крохотный размер скрывает впечатляющие возможности; но как второй осциллограф, возможно, будет лишним.

Рейтинг 8/10

Lumino от 4tronix

Компактная роботизированная платформа на базе вечно популярного Arduino. Лес Паундер наблюдает за восстанием машин.

Вкратце

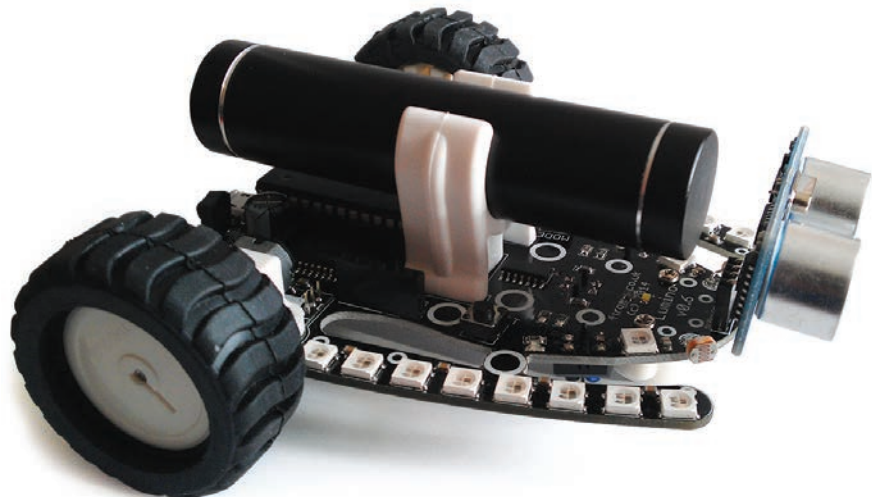
» Роботизированная платформа на базе популярной платформы Arduino, использует твердую печатную плату как шасси. См. также: Pi2Go.

Роботы захватывают власть над миром, но не так, как показывал Джеймс Кэмерон в классическом фильме «Терминатор». Вместо этого они уверенно завоевывают мир встраиваемых и физических вычислений. Ценным навыком стало умение программировать, но порой оно разочаровывает. Как бы крутые скитлы вы ни использовали для повтора строк, с точки зрения более юных хакеров вывод текста на экран выглядит довольно нудно. Зато робототехника — это увлекательная, чарующая форма программирования, ведь она даёт детям возможность по-настоящему управлять роботом, одновременно обучаясь принципам программирования при выполнении задачи, поставленной роботу.

Lumino — новейшая роботизированная платформа от 4tronix, и она основана на платформе Pi2Go той же компании, с той разницей, что Lumino вместо Raspberry Pi использует популярную плату Arduino.

Lumino следует тем же принципам построения, что и Pi2Go: ходовая часть является одной большой печатной платой, что придаёт роботу прочности. На борту размещаются компоненты, которые вместе и становятся роботом, причём можно выбрать из множества опций.

Например, Lumino поставляется с набором сверх-ярких «нео-пикселей» WS2812 — цветных светодиодов, расположенных по его периметру и управляемых платой WS2811. Светодиоды создают



» Lumino — исключительно компактная платформа с большим потенциалом для расширения, благодаря своему дизайну и деятельности сообщества Arduino.

отличный эффект и дают визуальную обратную связь от трёх датчиков слежения за линией на дне робота, позволяющих ему следовать намеченному курсу. Помимо этих датчиков, Lumino также располагает датчиком ультразвука, который выглядит как пара глаз спереди у робота. Эти «глаза» позволяют Lumino определять расстояние до объектов, встречающихся на его пути.

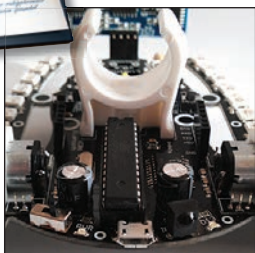
Кроме внушительного набора датчиков, у Lumino есть пара неординарно выбранных интерфейсов: Bluetooth и инфракрасный приёмник. Функция Bluetooth, как вы могли догадаться, может применяться для беспроводного взаимодействия с Lumino на коротких расстояниях, а инфракрасный приёмник позволяет управлять роботом при помощи пульта из комплекта.

контроль над двумя электромоторами, которые двигают Lumino вперёд.

В целом, Lumino хорошо продуман, и в нём предусмотрели место для аккумулятора для беспроводных операций. Выбор Arduino Uno (микроконтроллер ATMEGA328P-PU) позволит Lumino получить всё от богатого сообщества и множества его Arduino-проектов. Программирование робота — относительно лёгкое дело даже для новичков в Arduino, а более опытные Arduino-хакеры могут использовать Lumino в проектах посложнее.

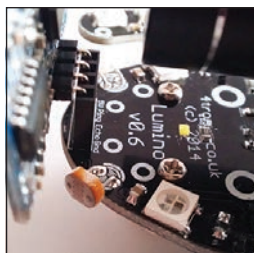
В качестве альтернативы робототехнике на Raspberry Pi, Lumino является отличной платформой для старта и может расти вместе с навыками пользователя. LXF

Свойства навскидку



Микроконтроллер

Выбор микроконтроллера ATMEGA328P-PU позволяет программировать Lumino через приложение для Arduino.



Сенсоры повсюду

В наборе много датчиков, в том числе ультразвуковые, датчики следования вдоль линии и фоторезисторы.

Расширяемость

В сердце Lumino лежит Arduino Uno-совместимое устройство, и его легко запрограммировать с помощью приложения Arduino и имеющегося USB-кабеля. Для Lumino есть несколько «скетчей» или «эскизов» — так называются примеры кода Arduino. Сюда входят и простой тест, проверяющий, что моторы работают как полагается, и цветовой шоу с использованием нео-пикселей. Набор включает несколько демонстраций, показывающих, как датчики следования вдоль линии и колёсные датчики работают, обеспечивая

LINUX FORMAT Вердикт

Lumino

Разработчик: 4tronix
Сайт: www.4tronix.co.uk/store
Цена: £60

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

» Прекрасная платформа для любителей роботов любого уровня. А контроллер ATMEGA328P-PU делает набор расширяемым.

Рейтинг 9/10

Geometry Wars 3

Иэн Дрансфилд сыграл в двуручную стрелялку, где стоит моргнуть — и умрёшь.

Вкратце

» Популярная классическая аркада получила развитие с 3D и режимом многопользовательской игры онлайн.

» Враги могут подкрасться к вам, обогнув игровую область, в стиле игры *Asteroids*.

Geometry Wars 3: Dimensions на первый взгляд выглядит очень простой: летай; пали по фигурам различной формы и различного поведения; не умирай. Банально. Но эта великолепная игра в конце концов съест столько вашего времени, сколько вы и представить боялись. Кривая сложности, которая сперва может ошарашить, станет посильной, когда вы привыкнете к базовым механизмам *Dimensions*. Процесс продолжится, и вы будете становиться всё искуснее в игре: ситуация, казавшаяся вчера безвыходной, завтра будет просто мелкой помехой.



Основные режимы используют знакомые принципы: набрать как можно больше баллов за заданное время/за одну жизнь/без выстрелов и так далее. Вам придётся то держать путь вокруг изрешечённого взрывами цилиндра, то спастись от вторжения Красных Стен Смерти в изрешечённом взрывами калейдоскопе. На каждом из 50 уровней можно получить от одной до трёх звёзд. Они необходимы для дальнейшего прохождения, поэтому повторное прохождение уровней — скорее требование, чем опция: хотя на уровнях и негде побродить, всё равно оказывается приятно вернуться и попробовать превзойти себя в коротких, но жёстких схватках.

По крайней мере — до тех пор, пока вы не дойдёте до сражений с боссами. Они не бессмертны, но используют дешёвые трюки вроде запугивания вас. Это разочаровывает: ведь обычная для *Geometry Wars* пальба придерживается более формальных рамок. Так или иначе, удивительно, что *Geometry Wars 3* оказалась

неплохой игрой: это неожиданный сиквел от относительно безвестной пробы пера от компании, которая была мертва не одно десятилетие. Но такая мощная задумка вытягивает игру из любых обстоятельств и заставляет порекомендовать её. Главное — не моргайте во время игры. **LXF**

LINUX FORMAT Вердикт

Geometry Wars 3: Dimensions

Разработчик: Aspyr Media
Сайт: www.gameagent.com
Цена: £ 12

Игровой процесс	8/10
Графика	8/10
Долгоиграемость	8/10
Оправданность цены	8/10

» Несмотря на угрожающую сложность, это потрясающее Linux-издание фантастической аркадной стрелялки.

Рейтинг **8/10**

Пуленепробиваемые SSL и TLS

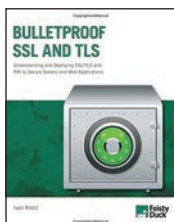
Джонни Бидвелл учится реализовывать защиту на четвёртом уровне.

Вкратце

» Всё, что вы хотели узнать, но боялись спросить о SSL/TLS.

» Вашему web-серверу следует быть пуленепробиваемым, как этот сейф.

Начав с краткого изложения основ — об уровнях OSI, генераторах случайных чисел, симметричном и ассиметричном шифровании, MitM-атаках — читатель погружается в подробный анализ протокола TLS. Это может отпугнуть новичков, но дальнейшие главы читаются на диво легко, хотя материал относительно сложный. Автор порой переходит на изложение от первого лица, что отдельные пуристы могут осудить, но это придаёт книге человечности. Частенько упоминаются действительные



атаки и приёмы, включая те, что обнаружил Сноуден, и рассказывается, как технологии улучшаются во избежание этих угроз. Книга вышла слишком рано и не затрагивает недавние уязвимости в Microsoft SChannel и, что, пожалуй, обиднее всего, атаку POODLE — последний гвоздь в крышку гроба SSLv3.

Вторая половина книги, опираясь на знания из предыдущего текста, даёт практические советы системным администраторам, для обеспечения максимально эффективного применения TLS и экосистемы PKI в сети. Скрупулёзно разобран жизненно важный пакет OpenSSL, с пошаговым руководством по созданию собственного CA (центра сертификации), и целая глава посвящена тестированию

возможностей серверов, где пакет установлен. Ещё вас ждут помощь в настройке *Apache* и *Nginx* и оптимистичный прогноз о будущем TLS. **LXF**

LINUX FORMAT Вердикт

Пуленепробиваемые SSL и TLS

Автор: Иван Ристич [Ivan Ristic]
Сайт: <http://bit.ly/SSLandTLS>
Цена: £ 40
Страниц: 530

» Очень информативно, хотя и не для начинающих. Текст настолько доступен, насколько это позволяет тема книги.

Рейтинг **8/10**



КОНКУРЕНЦИЯ

«Ваше слово, товарищ Android!»

Apple ослабила позиции Android.

Данные, приводимые исследовательской компанией Kantar Worldpanel, свидетельствуют: успешные продажи iPhone 6 позволили платформе iOS в четвертом квартале минувшего года, хотя и с минимальным перевесом, вернуться на верхнюю позицию среди мобильных операционных систем в США (два года назад в этой категории Android забрал место лидера у платформы Apple): на iOS пришлось 47,7% продаж смартфонов на американских рынках в IV квартале 2014 г., а рыночная доля Android составила 47,6%. Начиная с IV квартала 2012 г., Android в США впервые опустился на второе место.

Android предустановлен на значительно большем количестве устройств, чем iOS, но Apple обошла ОС от Google за счёт продаж немногочисленных моделей iPhone. Наибольший вклад в успех Apple внёс iPhone 6, продажи которого начались в сентябре. В течение IV квартала, включающего сезон праздничного шоппинга, этот смартфон

пользовался в США наибольшим спросом. Другой залог успеха Apple — широкий ценовой диапазон. iPhone 6 Plus с большим экраном и более старые, но и более дешёвые модели успешно конкурировали с устройствами high-end и среднего уровня на Android. За последние три месяца 2014 г. были проданы рекордные 74,5 млн iPhone 5 и iPhone 6.

Чтобы привлечь покупателей, выходящий на современный рынок новый смартфон должен обладать некой «изюминкой», чего нет у последних устройств на Android, считает Каролина Миланези [Carolina Milanese], руководитель отдела исследований Kantar. Продажи Samsung Galaxy 5, позиционировавшегося производителем в роли конкурента iPhone, были вялыми, а HTC, со своей флагманской моделью One M8, испытывает трудности из-за ослабления бренда. Сможет ли Galaxy 6, который Samsung представил 1 марта, перед началом Mobile World Congress, вернуть интерес потребителей?



» Есть ли у новой модели Samsung та изюминка, что способна вернуть внимание пользователей к Android?

В Европе доля Android снизилась в 2014 г. на 3,8%, до 66,1%, тогда как доля Apple поднялась на 6,2% — до 24,1%. Единственной европейской страной, где в прошлом году показатели Android на рынке выросли, была Италия. Кроме того, Apple укрепила свои позиции в Китае, увеличив в IV квартале 2014 г. долю на его рынке до 21,5 против 19% в аналогичном периоде годичной давности.

ХИТРЫЙ СПОНСОР

Данайцы, дары приносящие?

Microsoft инвестирует в Cyanogen.

Как сообщает The Wall Street Journal, Microsoft намерена инвестировать в проект Cyanogen около \$70 млн. Прямые инвестиции Microsoft в Cyanogen связаны с потенциалом проекта: вытолкнуть Google с рынка мобильных ОС, используя для этого кодовую базу самого Google. Альтернативная прошивка CyanogenMod довольно популярна среди тех пользователей Android, которым не хватает гибкости этой операционной системы, а инвестиции Microsoft обеспечат корпорации некое влияние на развитие проекта.

История с инвестициями Microsoft в Cyanogen перекликается с недавним заявлением CEO Cyanogen Кёрта Макмастера [Kirt McMaster]: «Cyanogen отберет Android у Google». Главная претензия Макмастера заключается в том, что Android навязчиво предлагает пользователю глубоко интегрированные с ним сервисы Google. Похоже, что Макмастер, являющийся пользователем iPad, забыл

об «открытой» природе iOS (недавно сообщество было разочаровано фактом разрыва Макмастером контракта с OnePlus в пользу Micromax, причём соответствующий email был отправлен с iPad, а не с устройства на Android).

Сравнение iOS и Android показывает значительные различия между стратегиями, которых придерживаются эти платформы. Apple блокирует практически всех сторонних игроков, лишь нескольких избранных допуская к интеграции со своей ОС. Только Twitter и Facebook получили доступ в мир iOS, а обмен контентом с Google+ или другими сервисами затруднён. В то же время Android является одной из наиболее открытых платформ. Если бы это было не так, в первую очередь не смог бы возникнуть сам проект Cyanogen: попробуйте, например, создать клоны iOS или Windows. В свою очередь, Android предлагает глубокую интеграцию с любым сервисом, который

установлен на устройстве — будь то Dropbox, ownCloud, Seafile или OneDrive.

Как будут развиваться события? Google потянет долю рынка, которая достанется Microsoft? И тогда уже Microsoft начнёт навязчиво предлагать свои сервисы, интегрированные с Cyanogen?



» Останутся ли пользователи верны Cyanogen, если одним из бенефициаров компании внезапно окажется Microsoft?

АМЕРИКАНСКИЙ ПЛАЦДАРМ

Первый шаг Xiaomi на рынок США

Руководители китайской компании провели пресс-конференцию в Сан-Франциско.

Крупнейший производитель смартфонов на базе Android, о котором, вероятно, большинство американцев и не слышали, делает свой первый шаг на рынок США. Компания Xiaomi, которая всего за 4 года прошла путь с нуля до крупнейшего игрока китайского рынка, запустит в этом году U.S.-версию своего сайта онлайн-продаж. Однако предложит компания пока не высокотехнологичные недорогие телефоны, которые, собственно, и создали репутацию Xiaomi, а широкий ассортимент аксессуаров: батареи, наушники и т.п. Это, впрочем, не означает, что выход телефонов Xiaomi на рынок США исключён.

«Для вывода [смартфонов и планшетов] на рынок необходимо приложить значительные усилия, — заявил на пресс-конференции руководителя компании в Сан-Франциско вице-президент Xiaomi по международным рынкам Уго Барра [Hugo Barra], ранее занимавший руководящую должность в Google. — Требуется наличие различных сертификатов, соглашения с перевозчиками, организация службы поддержки клиентов. Это невероятный объём работы, и мы должны держать правильный темп. Пока же мы начинаем

с простых продуктов». Конкретные сроки по началу продаж мобильных устройств не назывались. Журналистам были розданы образцы смартфона MiNote 2 с просьбой сообщить компании своё мнение об устройстве.

В Китае Xiaomi стала фаворитом благодаря наличию активной обратной связи с местными пользователями и постоянному стремлению к улучшению своей версии операционной системы Android. Пользователи голосуют по поводу необходимости добавления определённой опции, и каждую пятницу, в 17:00 по местному времени, одобренное обновление публикуется. Так, пятница 13 февраля была отмечена 225-м еженедельным обновлением.

Примерно раз в год Xiaomi выпускает новый телефон. Высокое качество и низкая цена — вот формула, которая позволила китайской компании переиграть Samsung и к концу 2014 г. овладеть самой большой долей китайского рынка. Зарубежные же амбиции компании пока были ограничены азиатскими странами. В Индии Xiaomi втянута с Ericsson в патентный спор, касающийся беспроводной технологии. На вопрос, не эта ли проблема



► Высокое качество и низкая цена — формула успеха недавнего стартапа, за 4 года ставшего крупнейшим игроком китайского рынка.

задерживает выход Xiaomi на рынок США, соучредитель и президент компании Бин Лин [Bin Lin] рассказал о количестве принадлежащих компании патентов, но на сам вопрос не ответил.

Руководство Xiaomi объясняет достигнутые успехи поддержкой своих пользователей (в компании их зовут «фанатами»). Да, в социальных медиа компанию поддерживают буквально миллионы. Посмотрим, трансформируется ли эта формула в успех на привередливом рынке США.

ПРОБЛЕМЫ ANDROID WEAR

«Умные» часы: Горе от ума

Стать лидером рынка среди платформ для носимых устройств у Google не вышло.

Время «умных» часов на Android ещё не стало? Android Wear усиленно продвигают, но партнёры Google — производители аппаратных средств, включая LG Electronics, Motorola Mobility и Samsung Electronics, в прошлом году поставили всего лишь 720 тыс. устройств.

Предполагалось, что модель Motorola Moto 360 обеспечит взлёт платформы Google для носимых устройств. Но по данным исследовательской компании Canalys, основная масса потребителей пока не спешит за покупкой. «Android Wear надо улучшить, — уверен аналитик Canalys Даниэль Матте [Daniel Matte], — и мы полагаем, что это сделают». По мнению Франсиско Херонимо [Francisco Jeronimo], директора компании IDC по исследованиям мобильных устройств для Европы, необходимо всестороннее улучшение Android Wear, в т.ч. интерфейса пользователя и времени автономной работы: «У меня немало мобильных устройств, и я нахожу интерфейс Android Wear трудным для изучения. А когда я, наконец, освоил его, он мне не понравился». Время

автономной работы удручает, и здесь нет простого решения. Спецчипсеты улучшат ситуацию, но размер «умных» часов не изменят: батарея остаётся маленькой. «На увеличение времени автономной работы нужен не один год», считает г-н Херонимо. Некоторые производители избрали неудачные дизайнерские решения. Так, для зарядки батареи модели от Samsung нужна подставка; но удобнее подключать зарядное устройство прямо к часам.

Главное же препятствие — не эти технические ограничения, а то, что в Google до сих пор не придумали мотивацию, зачем потребителю покупать «умные» часы именно с Android Wear.

С этими недостатками Android Wear не повторить успех Google на рынке смартфонов.

Конкурент Google, Pebble Technology, от своего основания в 2013 г. и до конца 2014-го выпустил 1 млн устройств. Постоянные обновления ПО, обилие приложений в App Store и осеннее снижение цен обусловили высокий уровень продаж в конце 2014 г. Но сейчас все следят за Apple и их Watch, которые намечено ввести на рынок в апреле.



► Ума у часов палата, а заряда хватит ненадолго.

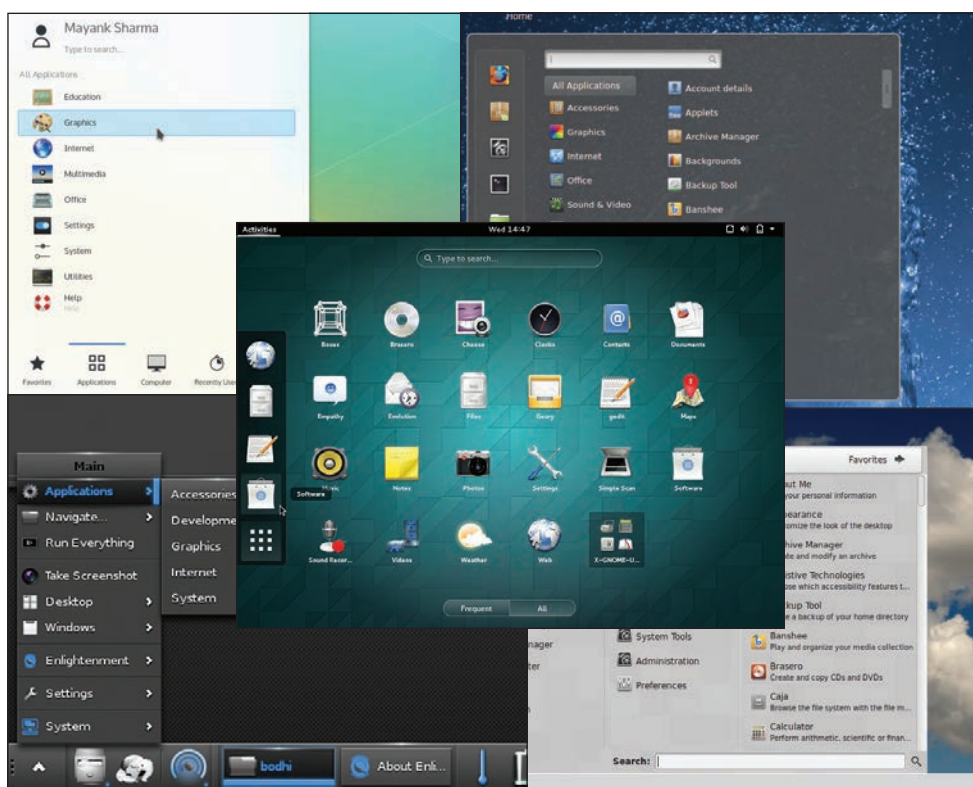
По словам Херонимо, судьба «умных» часов в руках Apple. «Неудача Apple убьёт эту категорию устройств: если не преуспеет Apple, то и ни одна другая компания». Но CEO Apple Тим Кук [Tim Cook] убеждён: «В Watch столько функций, что потом не представить жизни без них. Как Apple изменила рынки MP3-плееров, смартфонов и планшетов, так её Watch изменят рынок „умных“ часов». LXF

Сравнение

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

Рабочие столы Linux

Вы недовольны рабочим столом по умолчанию в своем дистрибутиве? **Маянк Шарма** составляет рейтинг некоторых популярных альтернатив.



Про наш тест...

Некоторые дистрибутивы поддерживают определенную среду рабочего стола, активно участвуя в ее разработке. Например, Fedora через своего корпоративного спонсора, Red Hat, оплачивает работу нескольких разработчиков Gnome. Подобным же образом многие постоянные разработчики KDE получают зарплату от openSUSE.

А коли так, мы решили, что тест должен проходить в нейтральной среде — и установили рабочие столы поверх основного дистрибутива Ubuntu, который идет только с Unity и не имеет ничего общего с разработкой какого бы то ни было другого рабочего стола. Однако мы также использовали родную среду, предлагаемую рабочим столом, дабы продемонстрировать работу его компонентов во всей полноте, а также сравнили их уровень настраиваемости, заодно обсудив имеющиеся в них инструменты настройки и все расширения — как от третьих сторон, так и поддерживаемые сообществом.

Для большинства пользователей Linux-ПК среда рабочего стола — главнейшее средство взаимодействия со своим дистрибутивом. Это подборка всех графических элементов, которые вы видите на экране компьютера, включая окна, панели инструментов, значки и т.д. Среда рабочего стола (Desktop Environment, DE) также включает менеджер окон [Window Manager], который отвечает за внешний вид окон в GUI.

Как и со всем прочим в Linux и открытом коде, пользователи избалованы богатым выбором DE. Возможность смены или переделки DE не менее важна, чем возможность смены приложений по умолчанию.

«Возможность смены DE не менее важна, чем возможность смены приложений по умолчанию.»

Большинство основных дистрибутивов официально поддерживают несколько рабочих столов. Fedora, Mageia, openSUSE поддерживают KDE, Gnome и ряд других. Затем есть дистрибутивы с официальной поддержкой лишь ограниченного числа рабочих столов, такие, как Ubuntu, который поддерживает только Unity, и Linux Mint, который предпочитает Cinnamon и Mate. Но это отнюдь не мешает

пользователю сменить официальный рабочий стол по умолчанию на другую опцию.

В нашем Сравнении мы рассмотрим некоторые самые популярные рабочие столы и их преимущества. Одно популярного рабочего стола в нашем списке нет: это Ubuntu Unity, который, хоть и относится к открытому коду, но лучше всего работает только на Ubuntu.

Наша подборка

- » Cinnamon
- » Enlightenment
- » Gnome Shell
- » KDE Plasma
- » Mate

Установка и поддержка дистрибутивов

Кто их поддерживает и как вы их получаете?

Вы можете перейти на другой рабочий стол, не меняя дистрибутив. Каковы бы ни были ваши цели и намерения, рабочий стол — это просто программа, и ее можно установить, как любую другую. Каждый рабочий стол в этом Сравнении поддерживается почти всеми дистрибутивами, то есть все они лишь требуют заглянуть в ваш менеджер пакетов.

Gnome — рабочий стол по умолчанию во многих популярных дистрибутивах.

Проекты Fedora, Mageia и openSUSE предлагают официально поддерживаемую устанавливаемую версию live CD/DVD на базе рабочего стола Gnome. Собственно говоря, Gnome был рабочим столом по умолчанию и в Ubuntu, пока там не начали использовать собственную оболочку. Однако Ubuntu по-прежнему пользуется основными библиотеками Gnome 3.

Следующий по популярности — KDE; openSUSE, Mageia и Fedora — вот некоторые из основных дистрибутивов, которые официально поддерживают этот рабочий стол и производят устанавливаемый образ live CD/DVD на базе KDE. Желая получить рабочий стол KDE на Ubuntu, поищите дистрибутив Kubuntu. Некоторые дистрибутивы, типа Linux Mint, включают несколько рабочих столов с разными пакетами, такими как *kde-standard* и *kde-full*.

DE Cinnamon — среда по умолчанию, предлагаемая

с дистрибутивом Linux Mint, который и возглавляет ее разработку. Этот рабочий стол имеется в официальных репозиториях Fedora, Mageia и Ubuntu, и вы можете установить его через их соответствующие пакеты. Mate пользуется спросом среди дистрибутивов, предназначенных для более старых компьютеров.

Enlightenment — один из старейших рабочих столов в нашем Сравнении, и тем не менее он не является выбором по умолчанию ни одного из популярных основных дистрибутивов. Его главным пропагандистом был Bodhi Linux, который недавно прикрыл лавочку, однако вы можете найти *Enlightenment* в официальных репозиториях практически любого дистрибутива.

Если вы установили несколько рабочих столов, переключаться между ними легко. Для этого просто выйдите из текущего рабочего стола. Затем поиграйте с кнопками менеджера приглашения-логина, и у одной из них обязательно выпадет список всех установленных рабочих столов. Выберите тот, который хотите использовать, и менеджер позволит вам туда войти.



» Проект Ubuntu имеет официально поддерживаемые версии для рабочих столов KDE, Gnome и Mate.

Вердикт

Cinnamon
★★★★★
Gnome
★★★★★
KDE
★★★★★
Mate
★★★★★
Enlightenment
★★★☆☆

» Большинство популярных дистрибутивов идут или с KDE, или с Gnome.

Приложения по умолчанию

Что вас ждет в укомплектованном пакете?

Обычно среда рабочего стола идет со своими основными приложениями. Например, обычная установка Gnome предлагает подборку из более чем двух десятков основных приложений практически для всех настольных задач, от управления изображениями,

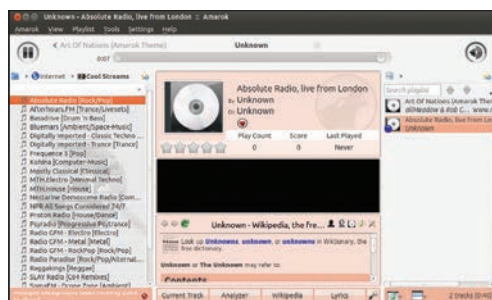
документами и контактами до воспроизведения музыки и видео.

Одна из особенностей Gnome 3 — более тесная интеграция с онлайн-сервисами через учетные записи онлайн [Gnome Online Accounts]. Это позволяет подписаться на такие сервисы, как Google Docs и Flickr, и делиться данными для использования оффлайн. В Gnome 3 есть также ряд приложений, использующих эти настроенные учетные записи онлайн, такие, как Gnome Contacts, что позволяет искать и редактировать ваши контакты, независимо от того, хранятся ли они локально или онлайн.

KDE располагает не меньшим количеством приложений. Его коллекции состоят из пакетов типа *KDE-Graphics*,

KDE-Admin и *KDE-Utilities*, и каждый из них включает соответствующие приложения: просмотрщики документов и изображений; полезные утилиты, такие как инструмент архивирования и калькулятор; и разнообразные инструменты для администрирования системы.

Другие рабочие столы не в состоянии тягаться с ними, и обычно включают только самые существенные приложения. Cinnamon использует многие приложения Gnome 3 со своими модификациями: например, менеджер файлов *Nemo* — ответвление *Nautilus* из Gnome. Подобным же образом рабочий стол Mate идет с рядом приложений, ответвившимися от Gnome 2, такими, как менеджер файлов *Caja*, текстовый редактор *Pluma* и программа просмотра изображений *Eye of Mate*. *Enlightenment* мало что предлагает. Это не вполне рабочий стол, и в нем нет собственных приложений.



» В отличие от былых дней, основные среды рабочего стола лучше взаимодействуют друг с другом, и приложения, разработанные для одного рабочего стола, запускаются на другом без всяких аномалий.

Вердикт

Gnome
★★★★★
KDE
★★★★★
Cinnamon
★★★★★
Mate
★★★★★
Enlightenment
★★★☆☆

» Gnome и KDE блистают, а Mate и Cinnamon предлагают лишь самое основное.

Внешний вид

Что обеспечивает лучшее впечатление от рабочего стола?

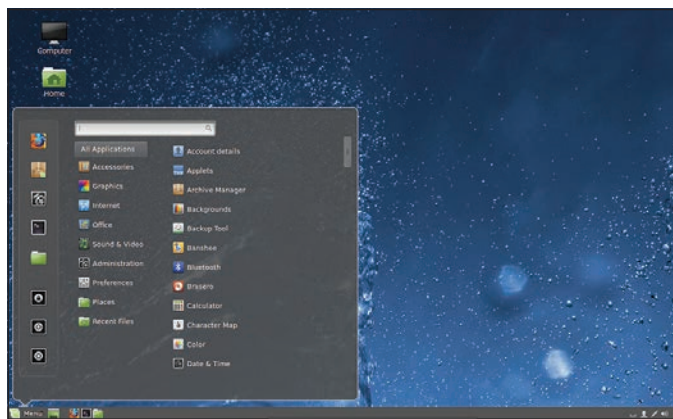
До появления Unity и Gnome 3 опции рабочего стола определялись дистрибутивами Ubuntu и Fedora. Однако сопутствующие этим DE релизы повели дизайн рабочих столов в весьма спорном новом направлении, перекраивая рабочий стол для следующего поколения компьютерных устройств, работающих

без привычного сочетания мыши с клавиатурой. В попытках соответствовать новым сенсорным устройствам обе среды рабочего стола лишились изрядной доли пользователей настольных ПК, которых вдруг поставили перед фактом необходимости изучать новые варианты общения со своими компьютерами.

Это противоречие вызвало к жизни Mate и Cinnamon. Но, как обычно и бывает с FOSS, с течением времени и Unity, и Gnome 3 обрели достаточную гибкость, чтобы вернуть себе часть своей бывлой аудитории. Однако они заметно отличаются друг от друга — и внешним видом, и способом работы с ними.

Cinnamon ★★★★★

Cinnamon — рабочий стол, который придерживается привычного представления о DE; появился он в порядке реакции на недовольство сообщества поведением Gnome 3 и Unity. Внешность у Cinnamon весьма традиционная, со значками на рабочем столе и панелью внизу, которая отображает уведомления вместе со списком открытых окон и меню Applications в нижнем левом углу экрана. Подобно Mate, меню Cinnamon Applications [Приложения] является обновлением стандартного меню с учетом новых технологий, и оно расширило привычную категоризованную структуру меню удобными функциями, позаимствованными из других сред — например, панели Favourite Apps. Кроме того, там предусматриваются виджеты, которые вам допускается размещать и на своей панели, и на рабочем столе. Предусмотрена также Exposé-подобная функция «горячий угол», которая представляет обзор открытых окон.



Enlightenment ★★★★★

Enlightenment Window Manager появился в 1997 г. и приятно отличался от других свежим взглядом на внешний вид рабочего стола, когда основными разновидностями такового были Gnome и KDE. По умолчанию на рабочем столе был Workspace Switcher [Переключатель рабочей области] вверху и панель внизу с программой запуска приложений. Можно щелкнуть левой кнопкой где угодно на рабочем столе, чтобы вывести программу запуска приложений, и перетаскивать значки на рабочем столе. Enlightenment предлагает новый механизм взаимодействия с окнами; например, для максимизации окна есть шесть разных опций. Последний Enlightenment весьма активно использует графику. На рабочий стол можно добавлять ряд виджетов (здесь их называют гаджетами). Очень радует, что, в отличие от других рабочих столов, графические эффекты на Enlightenment, вроде постепенно исчезающих меню, отлично работают со старым, не слишком мощным оборудованием.

Помощь и документация

Нужна рука, чтоб на нее опереться?

Все рабочие столы предлагают вполне адекватные возможности получения помощи и поддержки. Например, у Gnome имеется портал помощи (<http://help.gnome.org>) для пользователей, а также <http://wiki.gnome.org>, со страницами о разных проектах Gnome и рядом списков рассылки и каналов IRC.

В KDE тоже имеется обширная документация для всех категорий пользователей. Есть документация почти по каждому приложению KDE на <http://docs.kde.org>, а также руководства по приложениям

и подсказки в UserBase wiki. Кроме того, предусмотрена TechBase wiki для продвинутых пользователей, с руководством системного администратора. И опять же, для получения поддержки подпишитесь на один из списков рассылки или форумов, либо загляните в каналы IRC.

В Mate есть wiki с пошаговыми инструкциями по его установке для нескольких дистрибутивов, а также список приложений Gnome 2 и их аналогов в Mate. Пользователи, нуждающиеся в помощи, могут обратиться на официальные форумы

или канал IRC. Как ни странно, для самого Cinnamon официальной документации нет, хотя руководство пользователя Linux Mint посвящает ему отдельный раздел. У Enlightenment есть wiki с подробной информацией о его компонентах, и вы можете попросить о помощи в канале IRC или в пользовательском списке рассылки.

Многие проекты, подобно Gnome и KDE, также помогают связаться с их разработчиками, собирая свои блоги в специальных порталах Planet — <http://planetkde.org> и <http://planet.gnome.org>.

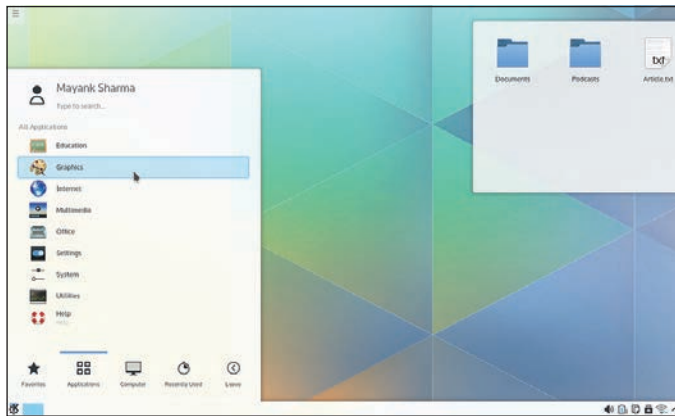
Вердикт

KDE ★★★★★
 Gnome ★★★★★
 Mate ★★★★★
 Cinnamon ★★★★★
 Enlightenment ★★★★★

» У всех рабочих столов имеется инфраструктура поддержки для пользователей.

Gnome 3 ★★★★★

Революционный интерфейс Gnome 3 по-прежнему требует привычки; мы заметили, что его приложения наилучшим образом выглядят в полноэкранном режиме при запуске внутри окон, где нет кнопки Minimise. Рабочий стол начинается с Activities Overview [Обзор действий], дающего доступ к установленным приложениям, с напоминающей запускатель приложений панелью Favourites [Избранное] для размещения часто используемых приложений. Workspace Switcher находится в правой части экрана и всегда перечисляет все добавочные рабочие области; переход на вторую рабочую область и добавление окон автоматически прибавят третью рабочую область. Вверху расположено окно поиска — через него ищутся совпадения введенных строк с приложениями и документами, которые хранятся локально или имеют ссылки на онлайн-сервисы. Упущения, раздражающие традиционалистов — нельзя заполнять рабочий стол значками и нет меню по щелчку правой кнопки.

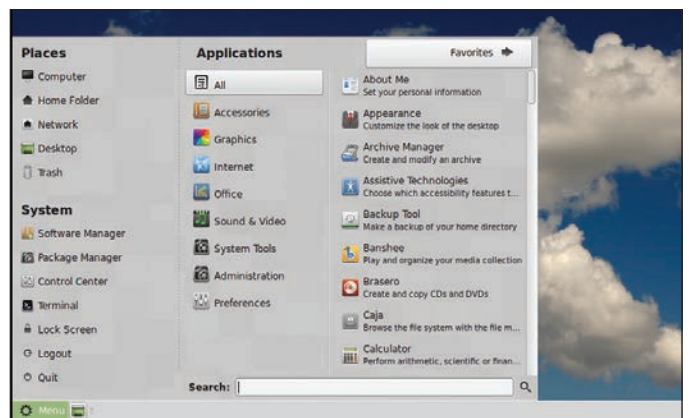


KDE ★★★★★

В противоположность Gnome, рабочий стол KDE — это гибкая оболочка, где индивидуальной настройке поддается абсолютно все. Структура по умолчанию и поведение рабочего стола и программы запуска приложений Kickoff покажутся знакомыми даже пользователям операционных систем, отличных от Linux. KDE идет с разными Views [Виды], призванными извлечь максимум пользы из наличного имущества рабочего стола для обычных экранов или нетбуков, и не принуждает пользователя к чему-то одному. Структура по умолчанию — Desktop View, который позволяет использовать виджеты рабочего стола, и можно также добавлять виджеты на панели. Большинство дистрибутивов вешают на рабочий стол виджет Folder View, чтобы показать содержимое папки в аккуратном небольшом окне, которое можно пристроить на любом месте своего рабочего стола. В новом релизе более плоские значки, и теперь его функция Activities удобнее для доступа и лучше настраивается.

Mate ★★★★★

Рабочий стол Mate является ответвлением Gnome 2, и до последней буквы копирует его внешний вид. В Linux Mint Mate выглядит иначе, и поставляется с панелью внизу и программой запуска приложений слева. По щелчку на программе запуска приложений появится трехпанельное меню приложений. На первой панели размещаются быстрые ссылки на инструменты Places and System, среди которых Package Manager, Control Center и т.п., а также опции выключения. На второй панели располагаются типовые программные категории, такие как Accessories [Стандартные принадлежности] и Office. Щелкните по любой из них, чтобы увидеть их содержимое в третьей панели. Многие пункты в этих панелях управляются через плагины, которые легко отключить в настройках панели. Можно создавать на рабочем столе значки и ссылки и размещать файлы и папки. Вы также можете добавить панель вверху и пристроить на нее апплеты, точно так же, как в Gnome 2.



Производительность

Как они влияют на время загрузки?

Важным критерием выбора рабочего стола является возраст совместимого с ним оборудования. Более новые рабочие столы требуют ускоренной графики и большого объема ОЗУ. На новых системах стоит выбирать рабочий стол общего назначения, например, Gnome 3, KDE или Cinnamon. Но на более старых машинах Mate и Enlightenment позволяют работать более ровно.

Gnome на старых машинах загружается медленнее и потребляет немало ресурсов. А вот KDE на тех же машинах будет

откликаться быстрее, и с каждым релизом становится экономнее в плане ресурсов. Но для более серьезной работы оба уместны только на машинах с ОЗУ от 2 ГБ.

Ключевое различие между Cinnamon и Mate в том, что первый извлекает преимущества современного «железа» для обеспечения шикарной графики, а Mate работает эффективнее на более старом оборудовании. Mate часто позиционируют как рабочий стол для пользователей, жаждущих производительности Cinnamon, но не имеющих на него ресурсов. На нашей

тестовой машине Mate загрузился почти вдвое быстрее, чем Gnome и KDE. Он также умудрился в среднем срезать почти 5 секунд по сравнению с Cinnamon, сохраняя практически тот же объем потребляемой памяти. В отличие от многих легковесных рабочих столов, Enlightenment имеет привлекательный внешний вид, достойный полноценного рабочего стола, и использует на порядок меньше ресурсов. В наших тестах на SparkyLinux Enlightenment загрузился почти на 7 секунд быстрее, чем рабочий стол Mate на том же дистрибутиве.

Вердикт

- Enlightenment ★★★★★
- Mate ★★★★★
- Cinnamon ★★★★★
- KDE ★★★★★
- Gnome ★★★★★

» Gnome нужна весьма современная система, чтобы он задействовал все свои мышцы.

Расширения и дополнения

Мне нужны расширения — много расширений!

Расширения сыграли важнейшую роль в принятии Gnome 3, и в этом проекте также имеется инновационный способ их установки и управления. Сайт Gnome Extensions перечисляет ряд расширений для установки недостающих функций. Их можно установить прямо с сайта всего парой щелчков мыши.

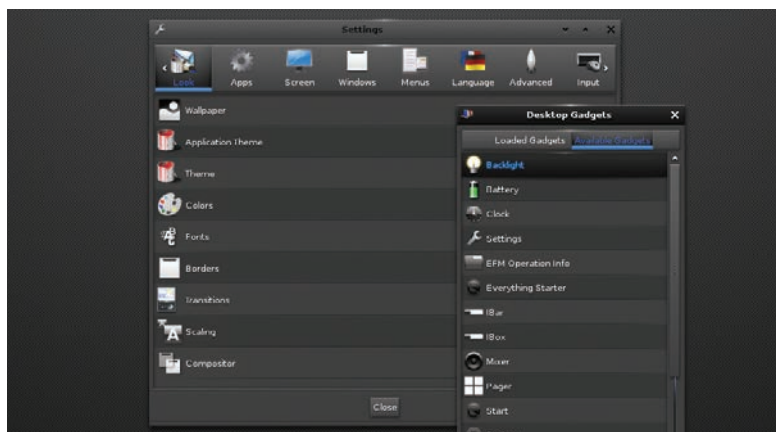
Популярны расширения, позволяющие использовать функции, которые были основными на рабочем столе Gnome 2 и которые облегчают переход на Gnome с Windows.

Одно из главных свойств KDE 4 — его расширяемость. Рабочий стол позволяет даже заменить стандартный запуск приложений Kickoff на его современную

версию Lancelot. Добавочные виджеты на жаргоне KDE именуются Plasmoids, и их вы можете найти десятки — от отображения лент RSS до автоматической загрузки изображений на сайт галереи.

Cinnamon поставляется с модулем Extensions в Control Panel. По умолчанию расширений в нем нет, и сперва надо переключиться на вкладку Available Extensions (онлайн), чтобы их скачать. Во вкладке перечислено почти два десятка расширений, включая несколько механизмов переключения приложений Alt+Tab, таких, как Coverflow App Switcher и 3D App Switcher. Среди других популярных расширений — Desktop Scroller и Wobbly Windows.

В Mate нет официальных расширений, но сообщество обогатило функциональность некоторых ключевых компонентов: например, расширение Caja-actions добавляет приложения в контекстное меню, а расширение sound-converter преобразует форматы аудиофайлов. Имеются также плагины для текстового редактора Pluma и программы просмотра изображений Eye of Mate, а также ряд апплетов панели.



» Enlightenment предлагает ряд виджетов, называя их гаджетами; можно поместить их в Shelf (здесь именуемый панелью инструментов) или добавить на рабочий стол.

Вердикт

- Gnome ★★★★★
- KDE ★★★★★
- Cinnamon ★★★★★
- Mate ★★★★★
- Enlightenment ★★★★★

» Как же всё меняется — масса дополнений Gnome 3 делают его очень гибким.

Настраиваемость

Можно ли их подрегулировать?

В Gnome есть основные опции настройки, включающие смену обоев, настройку учетных записей онлайн и основные настройки конфиденциальности. Если нужны расширенные настройки, потребуется сторонний инструмент *Gnome Tweak Tool*, который имеется в официальных репозиториях большинства дистрибутивов, использующих Gnome — например, Fedora. С помощью этого инструмента

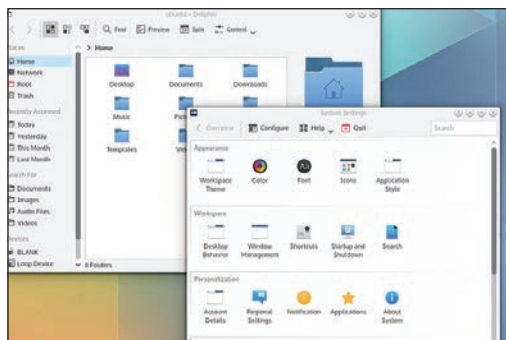
можно настроить внешний вид своего рабочего стола, отображение значков на рабочем столе, верхнюю панель, изменить поведение окон и рабочих областей, и предпринять многое другое.

Возможности настройки KDE буквально безграничны. Настройка KDE — это непрерывный процесс, а не разовое действие: рабочий стол создан для того, чтобы расти и изменяться вместе с вашими потребностями. В KDE имеется хорошо спланированный модуль System Settings. Настройки сгруппированы в две категории панели. С помощью опции Application Appearance можно выбрать тему для виджетов, а также воздействовать на отдельные элементы, такие как цвет, значки и шрифты. Обратите также внимание на опцию Desktop Effects, которая, как и предполагается

ее названием, позволяет активировать и настроить композиционные эффекты.

И Cinnamon, и Mate включают множество опций индивидуальной настройки в свои Центры управления [Control Center]. В Mate Control Panel позволяет влиять на стиль рабочего стола и на настройку оборудования и системных инструментов. Оба рабочих стола включают модуль Appearance, содержащий настройки для изменения вида рабочего стола, а в Cinnamon также имеется модуль Effects, где включается или отключается множество эффектов композитинга. Масса настроек размещается в разделе Preferences. С помощью модуля Panel, например, можно настроить структуру панели, переместить ее в другой угол экрана или автоматически скрывать ее, чтобы максимизировать пространство экрана на небольших устройствах.

Enlightenment тоже позволяет индивидуально настроить каждую деталь того, как он выглядит и работает. Его панель настройки просто переполнена разнообразными опциями.



» Некоторые дистрибутивы с KDE, скажем, openSUSE, включают собственный мастер настройки.

Вердикт

- Cinnamon ★★★★★
- KDE ★★★★★
- Enlightenment ★★★★★
- Mate ★★★★★
- Gnome ★★★★★

» Gnome 3 по очкам отстал от других, поскольку их опции настройки встроены.

Среды рабочего стола

Вердикт

Рабочий стол — дело личного вкуса. Далеко не все сразу забраковали Gnome 3, когда он выпустил свою совершенно новую версию. И не все сразу с восторгом приняли Cinnamon или Mate. Каждый рабочий стол разрабатывался с определенной целью, и соответствует определенному типу пользователей Linux.

Enlightenment среди наших рабочих столов самый загадочный: его красота и прелесть создаются в ущерб удобству. Лучший рабочий стол минус всякие побрякушки — это Mate. Но он не подходит как повседневный без сторонних приложений.

Если вам нужно, чтобы ваш рабочий стол был готов к использованию сразу, вы не найдете ничего лучше Gnome и его набора приложений по умолчанию. Но его эксцентричная структура уменьшает удобство в работе, и чтобы ваш рабочий стол стал действительно продуктивным, придется обратиться к стороннему инструменту индивидуальной настройки.

Если у вас индивидуальный стиль работы и вы хотите взять на себя контроль за структурой и функциями своего рабочего стола, то вам лучше брать KDE. Этот рабочий стол настолько гибок, что даже настраивается под внешность Gnome 3. Вот почему такие дистрибутивы, как openSUSE, Mageia, ROSA и Chakra, отличаются друг от друга, хотя все они предлагают рабочий стол KDE. Однако KDE — один из наименее дружелюбных к новичку рабочих столов, и все его опции настройки могут только отбить охоту работать с ним.

И мы объявляем победителем нашего Сравнения Cinnamon. Этот рабочий стол — одна из причин успеха Linux Mint, решившего пройти чуточку дальше, чтобы удовлетворить пользователей, недовольных новыми рабочими столами Gnome и Unity. Хотя Cinnamon не хватает индивидуальной настройки KDE, он предлагает адекватные



» Некоторые дистрибутивы поддерживают один рабочий стол и предлагают самое лучшее взаимодействие именно с ним.

готовые опции. И он интуитивен в использовании — по этой причине его предлагают в виде опции немало дистрибутивов, таких, как Mageia, Fedora и openSUSE. Фактически, за небольшим исключением, основные дистрибутивы официально поддерживают несколько рабочих столов. openSUSE, Fedora, Mageia поддерживают KDE, Gnome, Cinnamon и ряд других, и стоит попробовать несколько, чтобы узнать, который лучше подходит лично вам. **LXF**

Через месяц:
Языки
скриптов

«Cinnamon: этот рабочий стол — одна из причин успеха Linux Mint.»

I**Cinnamon** ★★★★★

Сайт: cinnamon.linuxmint.com Лицензия: GPL v2 Версия: 2.2
» Этот рабочий стол успешно соединяет старое с новым.

IV**Mate** ★★★★★

Сайт: www.mate-desktop.org Лицензия: GPL, LGPL Версия: 1.8
» Для любителей сохранять статус-кво.

II**KDE** ★★★★★

Сайт: www.kde.org Лицензия: GNU LGPL Версия: 5.1
» Рабочий стол для любителей индивидуальной настройки.

V**Enlightenment** ★★★★★

Сайт: www.enlightenment.org Лицензия: BSD Версия: e19
» Идеален для старых ПК, которым не по зубам DE-тяжеловесы.

III**Gnome** ★★★★★

Сайт: www.gnome.org Лицензия: GPL, LGPL Версия: 3.14
» Он смел — и непохож на других; к нему надо привыкать.

Обратная связь

Вы согласны с нами? Или используете рабочий стол, который мы упустили из виду? Напишите нам на lxformat@futurenet.com.

Рассмотрите также...

В средах рабочего стола, которые вы можете установить на свой любимый дистрибутив Linux, недостатка нет. Имеется еще Unity, который, правда, не очень-то поддерживается за пределами Ubuntu от Canonical.

До появления Mate пользователи, нуждающиеся в легковесной альтернативе популярным рабочим столам, обращались к Xfce или

LXDE, и когда появился Gnome 3, многие ушли на Xfce, по причине сходства этого рабочего стола с Gnome 2. Затем есть LXDE, разработанный для задействования минимума ресурсов и имеющий куда более простые инструменты, чем даже Xfce. И LXDE, и Xfce официально поддерживаются разновидностями дистрибутива Ubuntu, именуемыми, соответственно, Lubuntu и Xubuntu.

Еще более быстрый рабочий стол — ROX Desktop. Он основан на менеджере файлов ROX Filer и вдохновлен пользовательским интерфейсом RISC OS. Некоторые дистрибутивы также используют менеджер окон Openbox. Для тех, кому подавай нечто еще более эзотерическое, есть JWM, применяемый в Puppy Linux и изумительно быстро работающий на старом оборудовании.

ВОТ ОНА — ТЕХНИКА

2015



С восторгом ребенка, получившего новую игрушку, Нейл Ботвик демонстрирует вам, как пробовать новейшие программы Linux.

Мы часто слышим, что Linux чудесно подходит для старого оборудования, и так оно и есть. Linux позволяет многим пользователям продолжать использовать старые системы, которые в ином случае пришлось бы просто выбросить. Однако подобные утверждения в то же время служат Linux плохую службу, создавая впечатление, что это — операционная система для бедных, для оборудования, которое не в состоянии работать с самой свежей и самой крутой [кхм, — ред.] версией Windows.

Более неверного утверждения и выдумать нельзя: в Linux есть очень современные программы, хотя самые популярные дистрибутивы не всегда это отражают. Команды разработки дистрибутивов должны думать о совместимости

и стабильности, поэтому они чаще всего по умолчанию включают стабильные, проверенные программы. Даже такие популярные дистрибутивы, как Ubuntu, который использует пакеты Debian Testing и имеет репутацию довольно

«Это программы, способные появиться в вашем дистрибутиве на будущий год.»

передового, не применяют самых последних программ. Так что же вам делать, если вы жаждете гарцевать на краю бездны? Какие программы пока не предлагаются дистрибутивами, и как попробовать их самим?

На следующих страницах мы рассмотрим некоторые новые технологии, разработанные для Linux, и сделаем это отнюдь не в теории. Это

будут программы, вполне способные появиться в вашем любимом дистрибутиве на будущий год; ну, а мы покажем вам, как познакомиться с ними уже сейчас. Вы увидите, что готовит вам грядущее, поскольку мы научим вас их устанавли-

вать. Мы рассмотрим также потенциальные сопутствующие проблемы: в конце концов, дистрибутивы избегают самого последнего экспериментального кода не без причин, и мы вам расскажем, как

использовать его, не рискуя своей системой или душевным равновесием. Какой у вас дистрибутив — неважно, хотя с одними работать проще, чем с другими (например, как с Arch или Gentoo), но такие, как вездесущий Ubuntu (или какой-либо производный от него, например, Linux Mint), или openSUSE, или Fedora станут отличной базой для экспериментов.

Ядро Linux

Сделайте своей Linux ОС пересадку сердца.

Давайте начнем с самого сердца системы Linux: с ядра. Новые версии появляются каждый два месяца, а между ними выходят небольшие обновления. Так что ядро, предлагаемое вашим новым дистрибутивом, будет уже немного староватым, особенно если дистрибутив придерживается долгосрочных релизов ядра. Нужно ли вам суетиться, обновляя ядро? Обычно ответ на эти вопросы — «нет», но есть и уважительные причины обновить его до последней версии.

» **Безопасность:** Возможно, в той версии, которую вы используете, обнаружилась ошибка, которая влияет на безопасность или стабильность вашей системы. В таком случае будет выпущено промежуточное обновление с отладкой, и ваш дистрибутив предложит его через свою обычную систему обновлений.

» **Драйверы:** Если у вас новое оборудование, то, возможно, драйверы будут включены только в последующие версии вашего дистрибутива, так что вам придется самим компилировать свой собственный.

» **Функции:** Возможно, вам понадобится доступ к какой-то функции в более новой версии ядра. Например, файловая система `btfs` разрабатывается довольно быстро, и если вы ее используете, ядро посвежее будет очень неплохой идеей.

» **Заплатки:** Вам нужно будет добавлять в ядро заплатки, которые добавляют или изменяют функции.

» **Блеск:** Технари любят блестящие новые программы, и у вас способен возникнуть соблазн использовать более новую версию ядра просто потому, что вы это можете.

Посвящение

Применение версии ядра, отсутствующей в вашем дистрибутиве, означает необходимость скачать и скомпилировать ее самостоятельно. Среди пользователей Linux это считается своего рода посвящением, и определенно даст вам полное право утверждать, что вы собрали собственное ядро. Причем в этом процессе нет никакой мистики, и перекомпилировать ядро на самом деле гораздо безопаснее, чем компилировать из исходника что-либо другое. Дело в том, что ядро компилируется как отдельный файл в `/boot` и запускается вашим загрузчиком, но старое ядро по-прежнему там присутствует: новое — это всего лишь еще один файл с немного иным названием. Даже если вы ухитритесь скомпилировать незагружаемое ядро, вы сможете тут же выбрать старое ядро из меню загрузки и попробовать снова.

Хотя можно скомпилировать ядро от имени обычного пользователя и затем переключиться в `root`, чтобы его установить, всё же процесс будет проще, если вы перейдете в `root` с самого начала. Скачайте нужное вам ядро с <https://kernel.org> и распакуйте `tar`-архив в `/usr/src`. Если надо применить к ядру заплатку, тут самое время это сделать. Процесс применения заплаток зависит от каждого отдельного случая, так что следуйте инструкциям к заплатке.

Теперь надо настроить ядро. Это можно сделать прямо с нуля, однако будет проще начать с текущей настройки ядра. Некоторые дистрибутивы включают функцию, дающую доступ к текущей настройке ядра из `/proc/config.gz`. В этом случае вам нужно войти через `cd` в директорию вашего нового ядра и скопировать в файл настройки вот что:

```
cd /usr/src/linux-3.x.y
```

```
zcat /proc/config.gz >.config
```

Когда файл настройки будет на месте, настройте его с учетом новых опций. Это можно сделать по-разному, но простейший вариант обновления до более новой версии — запуск

```
make oldconfig
```

Вас уведомят обо всех различиях между текущей сохраненной настройкой и тем, что имеется в новом ядре. Обычно у вас будет четыре опции: `y`, `m`, `n` или `?`. Первая встраивает опцию в ядро, вторая компилирует ее как загружаемый модуль; вы, возможно, уже догадались, что `n` отключает опцию, а `?` выводит подсказку и затем снова задает вопрос. Другие опции предлагают графическую (более или менее) программу настройки.

```
make menuconfig
```

```
make xconfig
```

`Menuconfig` — программа на базе `ncurses`, и ее можно использовать через SSH или в консоли; `xconfig` открывает полно-

«Перекомпилировать ядро безопаснее, чем компилировать из исходника что-либо другое.»

стью графический инструмент. Используя любой из этих методов, вы сможете просматривать и выбирать опции и читать справочный текст. Можете также искать определенную опцию, нажав на `/` в `menuconfig` или `Ctrl+F` в `xconfig`. Затем нажмите на нужный вам пункт в `xconfig` или на число рядом с ним в `menuconfig`, чтобы перейти к данной настройке. После настройки можно скомпилировать и установить ядро и модуль таким образом:

```
make all
```

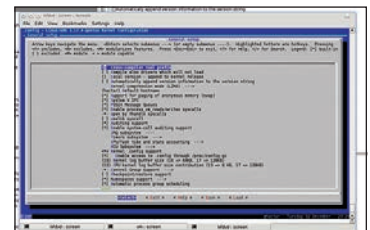
```
make modules_install
```

```
make install
```

Иногда нужная вам функция или драйвер имеется в текущем ядре, но не включена в сборку вашего дистрибутива. В подобном случае нужно не обновлять ядро, а перекомпилировать имеющееся.

Обновите `initrd`

Возможно, прежде чем запускать `grub-mkconfig` или `update-grub`, вам также придется создать новый `initrd`. Точная команда для этого зависит от вашего дистрибутива, однако этого можно избежать, скомпилировав всё, что вам нужно для монтирования раздела `root` — например, драйверы SATA и файловую систему, используемую на разделе `root` — в ядре, а не в виде модулей. У вас пропадет графический экран загрузки, но если вы работаете с самыми последними программами, вы всяко захотите видеть сообщения загрузчика.



» Когда доходит до настройки ядра, у вас есть выбор интерфейсов: установить удобный для работы с мышью `xconfig` и удобный для работы в оболочке `menuconfig`.

Wayland

Попробуйте систему окон, альтернативную устаревающей X11.

Один из самых обсуждаемых проектов, которого мы ждем не дождемся — это *Wayland*. Он должен заменить старую систему окон X, с ее переусложненной моделью клиент/сервер. *Wayland* находится в разработке уже пять лет, и дистрибутивы вроде Fedora неоднократно заверяли, что включают его в очередной релиз... но за этим следовало: «он пока не готов, но, возможно, появится в следующем релизе».

Wayland вообще-то уже доступен, хотя с ним ладят не все приложения. Вы можете установить его из пакета обычным способом или взять исходник с <http://wayland.freedesktop.org>, однако не исключено, что в вашем дистрибутиве *Wayland* уже установлен. Самые последние релизы Ubuntu и Fedora уж точно его установили. Однако сама по себе установка *Wayland* ничего не изменит. *Wayland* — это, главным образом, библиотека, а чтобы вы смогли заменить X, нужен компоновщик. Что же такое компоновщик? Это та самая программа, которая рисует ваш экран, занимается рендерингом окон в фоновом режиме и обновляет изображение по мере того, как объекты открываются, закрываются, изменяют размер и перемещаются. В современных системах компоновщик выполняет львиную долю работы, делая большую часть X ненужной. Это — одно из преимуществ *Wayland*; он выступает в роли простого интерфейса между программами и компоновщиком и между компоновщиком и ядром, которое обрабатывает

вводимые события, создавая более простую, более легковесную и более быструю систему, где каждая часть программного пакета работает со своими задачами, а *Wayland* обеспечивает их взаимосвязь.

Компоновщик проекта *Wayland* называется *Weston*, и его тоже надо устанавливать. Вы можете запустить сессию *Weston* в окне на X — достаточно скопировать в терминале `weston`; однако это не даст вам верной картины. Чтобы запустить полный рабочий стол *Wayland/Weston*, выйдите из X в виртуальную консоль и наберите

```
weston-launch
```

Делайте это как обычный пользователь, а не root. Перед вами откроется примитивный рабочий стол, где есть только значок терминала в правом верхнем углу. В *Wayland* есть примеры программ, запускаемых из терминала, например:

- » `weston-image` — просмотрщик изображений.
- » `weston-pdf` — просмотрщик PDF.
- » `weston-flower` — демонстрация графики.
- » `weston-gears` — запуск GLXgears, для сравнения с X.

Сделайте что-нибудь полезное

Демонстрационные программы хороши, но проку от них мало. Чтобы у вас был более функциональный рабочий стол *Weston*, вы должны отредактировать `~/.config/weston.ini`, который следует считать *Weston*-эквивалентом `.xinitrc`: он указывает, что запускать при загрузке рабочего стола. Он следует стандартному формату файлов INI — название раздела в квадратных скобках, за которым следует одна или более настроек для этого раздела, вот так:

```
[core]
modules=desktop-shell.so,xwayland.so

[shell]
background-image=/home/nelz/wallpaper.png
background-color=0xff000000
panel-color=0x90ffffff

[launcher]
icon=/usr/share/icons/hicolor/24x24/apps/chromium-browser.png
path=/usr/bin/chromium

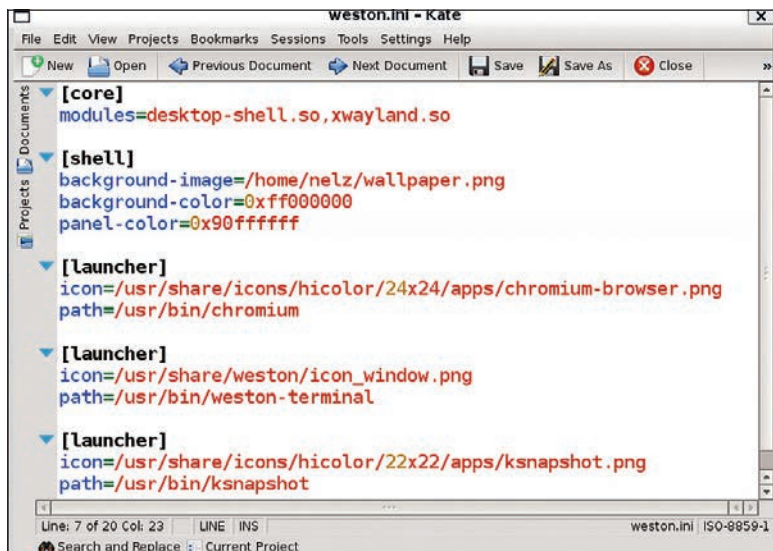
[launcher]
icon=/usr/share/weston/icon_window.png
path=/usr/bin/weston-terminal

[launcher]
icon=/usr/share/icons/hicolor/22x22/apps/ksnapshot.png
path=/usr/bin/ksnapshot
```

Раздел `core` загружает нужные вам модули; `xwayland` используется для запуска программ X на *Wayland*. Раздел `shell` задает основы рабочего стола: здесь мы настраиваем фон рабочего стола, цвет и прозрачность панели в верхней части экрана. Вы можете увидеть, что получилось, набрав `weston` в X терминале, чтобы открыть его в окне, но пока это мало что дает — давайте немного обогатим программу запуска:

```
[launcher]
icon=/usr/share/icons/hicolor/24x24/apps/chromium-browser.png
```

» На данном этапе, чтобы получить функциональный рабочий стол на *Weston*, нужно редактировать `weston.ini`. Поскольку программа еще на тестировании, удобные графические редакторы следует ожидать позднее.



Чем плох X?

Основная проблема X в том, что это — огромный набор программ, который пытается делать абсолютно всё для графической среды, хотя большую часть всего этого можно сделать где-то еще. Архитектура также означает, что всё должно проходить через X-сервер. Об открытии или закрытии окон клиенты сообщают не компоновщику, а X-серверу; сам же X-сервер не работает с подобными вещами,

а передает сообщение компоновщику, и ответ идет по не менее сложному маршруту. Подобным же образом, через X-сервер должны проходить события ввода, которые в наше время генерируются ядром.

Итак, эта сложная программа выступает в роли пейджера, к тому же не самого эффективного. Важно понять, что разработчики *Wayland* не ставят целью конкурировать с разработчиками X.org:

есть такие разработчики X.org, которые работают и на *Wayland*.

Проект *Wayland* является попыткой сделать инфраструктуру графического рабочего стола современной. Он занял куда больше времени, чем предполагалось (как это нередко бывает), однако он определенно добивается своего, и на него стоит обратить внимание.

Добавление репозиторийев

Менеджер пакетов любого дистрибутива позволяет добавлять внешние источники программных пакетов, называемые репозиториями. Дистрибутив даже может иметь собственные дополнительные репозитории, не включаемые по умолчанию, поскольку программы в них считаются нестабильными или экспериментальными. Обычно дистрибутивы предлагают список дополнительных ресурсов, например:

» **openSUSE:** https://en.opensuse.org/Additional_package_repositories

» **Fedora:** http://fedoraproject.org/wiki/Third_party_repositories

» **Ubuntu:** <https://launchpad.net/ubuntu>

» **Arch Linux:** <https://aur.archlinux.org>

» **Gentoo:** <http://wiki.gentoo.org/wiki/Layman> и <http://qpo.zugaina.org>

Launchpad от Ubuntu предлагает доступ к немалому числу репозиторийев, которые зачастую охватывают всего несколько пакетов, именуемых PPA (Personal Package Archive). Их можно добавить в ваш менеджер пакетов командой

```
sudo add-apt-repository ppa:user/ppa-name
```

Эти репозитории также способны работать с другими дистрибутивами, производными от Ubuntu, например, с Linux Mint, а часто и с прародителем Ubuntu — дистрибутивом Debian.

На сайтах проектов также бывают ссылки на пакеты для разных дистрибутивов, и обычно это — самые свежие пакеты. Но у вас в любом случае остается возможность заработать себе дополнительные очки умника, скомпилировав из исходника.

```
path=/usr/bin/chromium
[launcher]
icon=/usr/share/wayland/icon_window.png
path=/usr/bin/wayland-terminal
```

Когда у вас появятся готовые к запуску приложения, вы сможете добавить другие разделы, например,

```
[screensaver]
path=/usr/libexec/wayland-screensaver
timeout=600
```

для добавления скрин-сейвера и

```
[keyboard]
keymap_model=pc105
keymap_layout=gb
```

для настройки клавиатуры. Именно это ваш дистрибутив настраивает по умолчанию с помощью X и предлагает графические возможности изменить эти настройки. Так будет и с Wayland, когда тот станет популярным, но сегодня вам придется отредактировать файл INI. Полный формат объясняется на tap-странице `weston.ini`, там можно много чего настроить, но для начала хватит и этого.

Gnome на Wayland

Wayland и X мирно сосуществуют в одной системе, так что устанавливайте Wayland и экспериментируйте с ним, сохранив возможность при желании вернуться к рабочему столу на X. То есть можно совершенно безопасно попробовать Wayland на своей основной системе (ну, не опаснее, чем знакомство с любой новинкой), но если вы просто хотите разок попробовать с ним поработать, можете сделать это на Gnome из среды Fedora 21 live, как описано дальше в этой статье.

А можно ли запустить имеющиеся рабочие столы на Wayland? И в Gnome, и в KDE поддержка Wayland есть, причем у Gnome имеется преимущество — на данный момент, поскольку в Qt5 поддержка Wayland лучше. При правильной установке сессии и библиотек Wayland вы можете выбрать Gnome на Wayland в качестве опции, заходя в систему с помощью GDM. Проще всего это сделать в Fedora 21 (на LXF DVD этого месяца): поскольку файлы Wayland устанавливаются по умолчанию, остается только установить пакет `gnome-session-wayland-session`, чтобы добавить опцию логина для использования Wayland. Вы даже можете запустить его из среды live-диска, выйдя из системы и снова зайдя в нее. Не выбирайте опцию Live User: она вернет вас в X. Вместо этого нажмите на пункт Not listed, введите имя пользователя `liveuser`, затем щелкните по небольшому значку слева от кнопки Next, где вы сможете выбрать Gnome на Wayland.

Вернемся к началу создания `weston.ini`: мы добавили строку модулей для загрузки двух модулей. Первым был `desktop-shell`, стандартный рабочий стол Weston (есть еще `tablet-shell` для

устройств с сенсорной панелью), и мы также загрузили `wayland`. Эти модули позволяют клиентам X работать на рабочем столе Weston, а значит, вы сможете запускать большую часть своих программ на Wayland, не дожидаясь их портирования.

Альтернатива: Mir

Говоря о новых серверах отображения и Wayland, нельзя не упомянуть Mir. Этот альтернативный сервер отображения, разрабатываемый Canonical, имеет много общего с Wayland и заимствует некоторые концепции у Android. Единственным вариантом рабочего стола, совместимого с Mir, сейчас является Unity 8, будущая версия рабочего стола Ubuntu. Другие разработчики, например,

«Wayland и X сосуществуют на одной системе; устанавливайте и экспериментируйте.»

Xubuntu, уже подумывают о переходе на Mir, но пока никто не решил. Wayland для обработки событий ввода использует среду ядра Linux evdev, а Mir применяет методы Android, что делает Mir потенциально более пригодным для употребления вне настольных ПК, поскольку Unity начинает оправдывать свое название. На сегодняшний день, если вы хотите попробовать новый движок графики на любом дистрибутиве, лучше выбрать Wayland.



» Можете опробовать Wayland уже сейчас, загрузившись в Fedora 21 с нашего диска и войдя в систему с опцией «Gnome на Wayland».

ФС нового поколения

Файловые системы бурно развиваются — установите самую новую.

Файловые системы — та область Linux, где в последнее время наблюдается бурное развитие. И на то есть две очень весомых причины: появление так называемых файловых систем «нового поколения» и широкое применение твердотельных жестких дисков, чьи достоинства и недостатки так сильно отличаются от привычных вращающихся дисков. В былые времена вы делили диск на разделы так, как считали удобным, а затем устанавливали на каждый раздел одну файловую систему. Файловые системы развиваются: ext2 приобрела журнал и превратилась в ext3, и, в свою очередь, ext3 превратилась в ext4. И хотя у ext4 есть масса преимуществ по сравнению с предшественниками, всё же она по-прежнему следует правилу одной файловой системы на один раздел.

Затем были разработаны менеджеры томов, типа LVM, который допускает разделение одного крупного раздела на виртуальные разделы, или логические тома, что придало системе куда большую гибкость. Нам случалось запросто добавлять, удалять и изменять размер этих томов, хотя с файловыми системами

по-прежнему приходилось работать отдельно. Добавление дисков с использованием RAID навесило еще один уровень работы. Предполагалось, что для работы с файловыми системами в соответствии с изменяющимся требованиями хранения необходимо три набора инструментов: *mdadm*, RAID и инструментарий ext4. Были доступны и другие файловые системы, в частности, ReiserFS и XFS, но у них были те же ограничения.

Есть еще одна файловая система, которая находится в ядре и предлагает набор функций, похожий на ZFS; она называется *btrfs* (кстати, способов произнести *btrfs* больше, чем текстовых редакторов в Debian). Проект был запущен Oracle, теми же людьми, которые купили Sun и открыли код ZFS. Поскольку *btrfs* находится в основном ядре, не будет никаких проблем с запуском ее на корневой файловой системе. *Grub* даже может загрузить ядро из файловой системы *btrfs*, хотя это становится менее уместно с системами UEFI, которым сначала нужна на диске файловая система FAT, пригодная для /boot.

Разработка *btrfs* идет семимильными шагами, и мы бы рекомендовали использовать самое свежее ядро, чтобы у вас был более зрелый код. Также потребуется установить пакет *btrfs-tools*, содержащий инструменты для пространства пользователя. Чтобы создать файловую систему *btrfs* на одном разделе, запустите

```
mkfs.btrfs /dev/sdb5
```

Желая создать массив RAID 1, скомандуйте

```
mkfs.btrfs --data raid1 /dev/sda5 /dev/sdb5
```

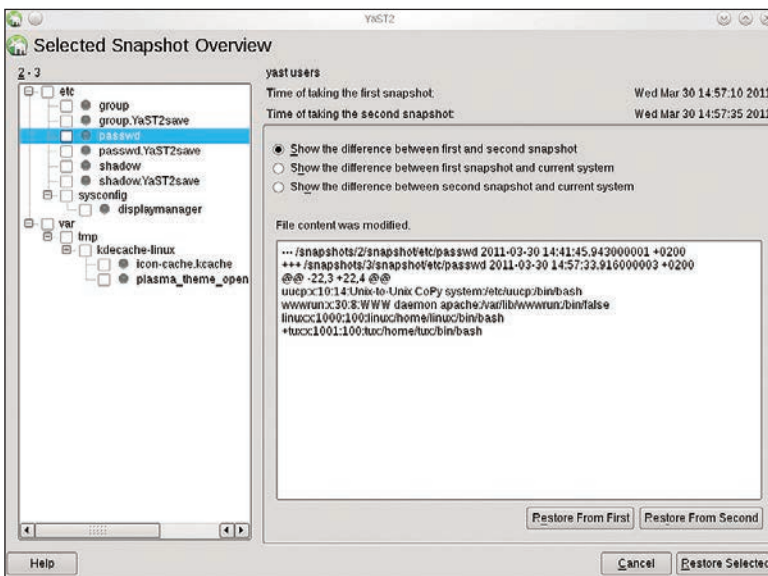
Btrfs обрабатывает RAID1 способом, отличным от классической модели копирования всех данных на все диски: она хранит две копии каждого блока данных, на разных дисках. В массиве из двух дисков это то же, что и обычный RAID1, но если вы добавите третий диск, у вас будет больше места для хранения, тогда как классический RAID1 предлагает тот же объем, но больший резерв. Два жестких диска емкостью 2 ТБ в обоих случаях дадут вам 2 ТБ, но добавьте третий в массив *btrfs* RAID 1 — и у вас уже 3 ТБ.

Применяем новую файловую систему

Создав свою новую файловую систему, смонтируйте ее обычным способом:

```
mount /dev/sdb5 /mnt/somewhere
```

Это работает для обоих приведенных выше примеров, и чтобы смонтировать массив RAID, вы указываете любое из его устройств. Теперь вы можете создавать подтома, которые будут действовать, как разные файловые системы, и могут иметь



► Мгновенные снимки системы — одно из множества улучшений, предлагаемых новым поколением файловых систем, подобных *btrfs*, и оно с успехом используется в инструменте резервного копирования *SUSE Snapper*.

Родончальник

Взгляд на файловые системы сильно изменился после того, как Sun представили ZFS, подлинную файловую систему «нового поколения», которая включала функции управления RAID и томами в самое себя. Вы могли установить в компьютер один или более дисков, дать ZFS команду сделать из них пул и затем насоздать в этом пуле тома. Ничего не надо было форматировать, потому что менеджером томов тоже была файловая система;

вы просто говорили ей, что вам нужно и какого размера. Если впоследствии вы меняли свое решение, вы просто при необходимости добавляли, изменяли размер или удаляли тома, без всяких проблем.

Когда Oracle открыл код ZFS, появился проект ZFS в Linux, и сейчас это уже весьма зрелая файловая система. Главный ее недостаток — несовместимость ее лицензии с GPL, то есть ее нельзя

включить в ядро, а надо устанавливать в виде отдельных модулей. Это не так уж страшно, потому что для большинства дистрибутивов модули имеются, но всё же делает перевод вашей корневой файловой системы на ZFS сложнее, чем он мог бы быть в ином случае. Еще один недостаток в том, что открытый код ZFS не является самым последним: в нем нет некоторых функций из более новых версий ZFS, например, шифрования.

Эксперименты с файловыми системами

Мы не предлагаем вам переформатировать ваш жесткий диск, чтобы познакомиться с экспериментальными файловыми системами, ведь есть и другие возможности: например, использовать внешний диск или второй внутренний диск, если они у вас имеются. Однако если вы не можете похвастаться наличием запасного диска, но зато у вас достаточно места на имеющемся, вы можете изменить размер существующих разделов, выделив себе место для экспериментов, или использовать устройство

обратной связи. Это позволит вам использовать большой файл как виртуальный диск, примерно так, как это делают виртуальные машины:

```
dd if=/dev/zero of=somefile bs=1 count=1 seek=10G
sudo losetup /dev/loop0 somefile
```

Первая команда создает пустой файл заданного размера. Вторая команда создает устройство обратной связи в **/dev/loop0**. Вы не должны пытаться создать устройство, которое уже используется. Если вы сомневаетесь, используйте для **losetup**

опцию **-f**, чтобы она выбрала первое имеющееся в наличии устройство, и затем **-l**, для просмотра сделанного выбора.

```
sudo losetup -f somefile
sudo losetup -l
```

Теперь можете использовать **/dev/loop0** так, как будто это раздел на настоящем диске, и экспериментировать, сколько душе угодно. Однако не забывайте, что вы всё же будете работать через настоящую файловую систему диска.

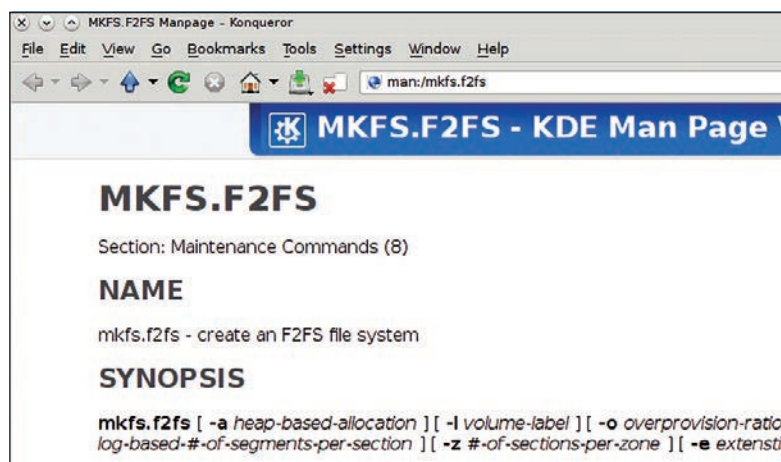
разные свойства, например, квоты или сжатие. И ZFS, и btrfs вычисляют контрольную сумму всех хранимых данных. Это слегка снижает производительность, но зато обеспечит намного большую безопасность данных, особенно при использовании RAID. Обе файловые системы могут определить и молча устранить повреждение данных (продублировав нормальную копию в RAID). Обе они являются файловыми системами copy-on-write [см. «Файловые системы: Новое поколение», LXF193, стр. 50], а значит, поддерживают моментальные снимки системы. Моментальный снимок подтома не занимает места на диске и создается почти мгновенно. Место он начинает занимать, когда вы вносите изменения в исходный подтом, а до тех пор хранится только одна копия данных. Это позволяет вам в любую минуту вернуться к тому состоянию, когда вы делали снимок — например, для восстановления удаленных файлов или для возврата к более старой версии того, над чем вы работаете.

Удобный для Flash

SSD становятся всё дешевле и популярнее (и мы уверены, что это не совпадение), хотя многие переживают насчет продолжительности жизни информации на твердотельных устройствах. В отличие от их более дешевых родичей, флэш-памяти на картах и USB-брелках, которые и сами уже способны вмещать куда больше записанной информации, чем раньше, в SSD есть хитроумные контроллеры, которые усредняют использование ячеек памяти (т.н. «выравнивание износа»), так что предполагаемый срок жизни подобных устройств часто больше, чем у вращающихся дисков.

Иными словами, SSD очень отличаются от магнитных дисков и предъявляют другие требования к файловым системам. Традиционные файловые системы оптимизированы для вращающихся дисков, на которых они применялись многие годы, но мы становимся свидетелями появления файловых систем, оптимизированных для флэш-памяти. Мы только что рассмотрели btrfs, где есть опции для использования на SSD, причем некоторые из них включаются автоматически при определении подобного диска. Есть еще одна файловая система, разработанная исключительно для использования на твердотельных жестких дисках: она называется F2FS (Flash Friendly File System).

Одна из проблем, с которыми могут столкнуться SSD — это потеря производительности со временем, потому что они терпят своего рода SSD-эквивалент фрагментации по мере удалений и записи данных на диск. Функция ядра TRIM работает с этой проблемой, но отчасти снижает производительность (поэтому желательно запускать ее из *cron*, а не в виде текущей автоматической операции TRIM). Однако F2FS использует другой подход для «уборки за собой». Эта файловая система была добавлена в ядро менее двух лет назад, но всё еще не считается стабильной, так что вполне соответствует другим программам здесь.



► Есть множество опций при форматировании SSD с помощью F2FS, но даже опции по умолчанию будут лучше для вашего флэш-диска, чем файловая система, разработанная для «ржавых вращающихся» дисков.

Чтобы активировать F2FS на вашем компьютере, нужно пройти две стадии. Первая — ядро надо скомпилировать с установленной опцией CONFIG_F2FS, либо встроенной, либо в виде модуля. Если в вашей системе существует файл **/proc/config.gz** (это зависит от опциональной настройки в ядре), вы можете проверить, настроена ли эта или любая другая опция, с помощью

```
zgrep F2FS /proc/config.gz
```

В ином случае можно проверить наличие модуля командой **modinfo** —

```
modinfo f2fs
```

и проверить, встроен ли модуль в ядро, с помощью следующей команды:

```
grep f2fs /lib/modules/kernel-version/modules.builtin
```

«Еще одна ФС разработана исключительно для твердотельных жестких дисков: F2FS.»

Если его нет в вашем ядре, скомпилируйте ядро, как описано ранее (см. стр. 31). Желая перевести свой раздел **root** на F2FS, вы должны встроить F2FS в свое ядро, а не в виде модуля. Следующая стадия — установить инструменты пространства пользователя, которые обычно именуются **f2fs-tools**, через менеджер пакетов своего дистрибутива. После этого создайте файловую систему **f2fs** командой

```
sudo mkfs.f2fs -l LABEL /dev/sdXN
```

Контейнеры Systemd

Попробуйте использовать несколько изолированных систем.

Контейнеры — это недавнее дополнение к ядру Linux. Они являются альтернативой виртуализации, если вы хотите запустить более одной системы Linux на том же хосте. В отличие от полной виртуализации, например, *Qemu*, *VirtualBox* и *VMware*, когда программа полностью эмулирует компьютер, контейнеры используют возможности ОС-хоста. Это означает, что в контейнере вы можете запустить только Linux — в отличие от полной виртуальной машины, где вы можете запустить что угодно: но в то же время это меньше нагружает ресурсы, поскольку контейнеры делают имеющиеся ресурсы, а не забирают себе часть ОЗУ и места на диске как только их запустят, независимо от реальной в них потребности. Контейнеры работают в собственном пространстве имен и используют функцию `sgroups` ядра Linux, чтобы обеспечить работу в изоляции от системы хоста.

Контейнеры называют «chroot на стероидах», и они действительно похожи на chroot, но куда более мощные и гибкие. Есть ряд способов применения контейнеров — Docker уже неоднократно упоминался в заголовках, и недавно ему было посвящено руководство в *Linux Format*, но мы рассмотрим работу с контейнером через *systemd-nspawn*. Это означает, что вам нужен дистрибутив, работающий с *Systemd*, что в наши дни является нормой, но вам он также понадобится и для Docker.

Запуск контейнера

Контейнер работает или с директорией, или с образом диска, содержащем операционную систему. Вы можете указать или одно, или другое, но не оба. Если не указано ничего, *systemd-nspawn* использует текущую директорию. Это означает, что вы можете соединиться с жестким диском с другого компьютера и запустить на нем ОС Linux, либо смонтировав его раздел `root` и вызвав

```
systemd-nspawn --directory /mount/point
либо указав путь к диску или к его образу и запустив
systemd-nspawn --image /dev/sdb
systemd-nspawn --image diskfile.img
```

Диск или образ диска должны содержать таблицу разделов GPT с обнаруживаемым разделом `root`. Чтобы защититься от попыток использования дерева файлов не-Linux, перед запуском контейнера *systemd-nspawn* проверит наличие в дереве контейнера `/usr/lib/os-release` или `/etc/os-release`.

Одно из ограничений chroot заключается в том, что он не очень удобен для изоляции ОС. Перед запуском chroot вам придется самим создавать всякие системные директории типа `/dev`, `/proc` и `/sys`, доступные через связанное монтирование [`bind mount`]. А вот *systemd-nspawn* позаботится обо всем автоматически, и директории монтируются только на чтение [`read-only`], чтобы работающие в вашем контейнере программы не смогли повлиять на ОС вашего хоста. Если вам нужно, чтобы какие-то директории были доступны внутри контейнера, можете указать их в командной строке, разными способами:

```
systemd-nspawn --directory /mount/point --bind=/mnt/important
systemd-nspawn --directory /mount/point --bind-ro=/mnt/important:important
```

Первая команда делает целевую директорию доступной в той же точке в контейнере, а вторая указывает два пути, где второй является путем внутри контейнера. Второй пример также иллюстрирует использование монтирования `read-only`. В командной строке можно использовать несколько вызовов `--bind`.

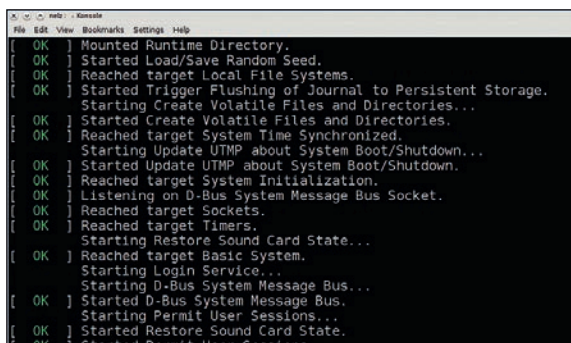
Загрузка внутри контейнера

Как и с chroot, запуск *systemd-nspawn* без аргументов запускает оболочку; по умолчанию это *Bash*. Опция, которая делает контейнер действительно интересными — `--boot`. Если она использована, то бинарник `init` в файловой системе `root` работает эффективно, загружая гостевую ОС в контейнере. Вы даже можете вызвать ее из скрипта загрузки, чтобы ваш контейнер запускался автоматически. Если вам потом нужно будет войти в контейнер, используйте команду `machinectl`. Без аргументов она перечислит все работающие контейнеры, а откроет сессию терминала в одном из них как

```
sudo machinectl login имя-контейнера
```

Systemd-nspawn именовывает контейнер по присвоенной ему директории, но вы можете использовать `--machine`, чтобы указать что-то более удобное.

➤ Похоже на обычный экран загрузки Linux, но присмотритесь внимательнее — и вы увидите, что он работает в терминале X и загружен в контейнере.



Заполнение директории контейнера

Мы рассказали вам обо всем, кроме того, как изначально поместить в контейнер ОС. Можно скопировать содержимое текущей инсталляции, или использовать образ диска, или начать с нуля. В Debian, Fedora и Arch есть команды для установки минимальной системы в директорию. Выберите одну из этих:

```
yum -y --releasever=21 --nogpg --installroot=~/  
mycontainer --disablerepo="*" --enablerepo=fedora install systemd passwd yum  
fedora-release vim-minimal  
debootstrap --arch=amd64 unstable ~/mycontainer  
pacstrap -c -d ~/mycontainer base
```

При запуске одной из этих команд вы должны работать в том же дистрибутиве, но после установки вы можете загружать контейнер из любого подходящего дистрибутива Linux (то есть того, который работает с *Systemd*) с помощью

```
sudo systemd-nspawn --boot --machine MySystem  
--directory ~/mycontainer
```

Новые технологии

Создайте тестовую среду и учитесь на практике.

Программы, которые мы здесь рассматриваем, по определению являются самыми передовыми. У вас они могут сработать или не сработать, привести к нестабильности в работе системы или конфликтовать с имеющимися программами. Или, наоборот, они могут работать хорошо и радикально улучшить вашу жизнь. Вы этого не узнаете, пока не попробуете на практике. Некоторых программ может не оказаться в репозиториях вашего дистрибутива, и вы станете не только тем, кто их испытывает, но и тем, кто их устанавливает. И всё это предлагается со стандартной гарантией открытого кода: если они приведут к поломке вашей системы, вы получите назад свои деньги и сохраните обломки!

Установка на вашу систему экспериментальных и неопробованных программ может быть удачным решением, только если вы не боитесь риска стереть и заново установить свой дистрибутив, если что-то пойдет не так. Есть несколько других опций:

» Проводить свои тесты на виртуальной машине. Она отлично подходит для тестирования дистрибутивов, но не так хорошо для программ, требующих настоящего графического оборудования и устройств хранения.

» Использовать другой компьютер. Это неплохая идея, при условии, что он будет достаточно мощным. Вам, возможно, придется компилировать некоторые программы из исходника, а на старой системе это отнюдь не сахар. Кроме того, старое оборудование не даст вам полного представления о работе таких программ, как, например, *Wayland*.

» Лучший вариант — двойная загрузка системы, которую мы более подробно обсудим ниже.

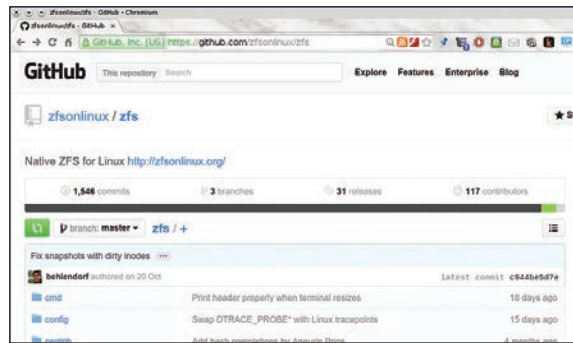
Двойная загрузка

Использование двойной загрузки — самый безопасный и при этом наиболее реалистичный способ создания тестовой среды. Установка еще одного дистрибутива вместе с вашей стандартной средой означает возможность сравнить производительность на одном и том же оборудовании. Программы установки большинства дистрибутивов имеют опцию изменения размера существующего раздела для установки вместе с ним другого. Имея отдельный раздел *home*, его легко поделить между двумя дистрибутивами и делать всё необходимое в любой среде.

Какой дистрибутив выбрать? Да хоть бы и тот, который является вашей основной системой: его вы хорошо знаете, и он делает сравнение более показательным. Однако в вашем дистрибутиве может не быть репозитория с новыми программами, так что вам, возможно, придется заняться компиляцией из исходника. Если вы хотите использовать другой дистрибутив, хорошим выбором будет Arch Linux. В нем есть пакеты почти для всего, и, что крайне важно при работе с новейшими программами, Arch Linux wiki предоставит отличную документацию.

Мы упомянули дополнительные репозитории. Хотя некоторые передовые программы имеются в стандартных репозиториях дистрибутивов, в официальных пакетах они всё же более склонны к стабильности. Это означает, что зачастую для установки прекомпилированных пакетов придется добавлять один-другой дополнительный репозиторий.

В некоторых случаях доступного двоичного пакета может не оказаться, или вы решите использовать версию поновее. Тогда



скачайте исходник и самостоятельно скомпилируйте его. Нередко это означает скачать tar-архив, распаковать его и следовать инструкциям в файле *Readme* или *Install*. Для некоторых программ нужно будет скачать исходник с GitHub. Скачивание с *git* отличается простотой и имеет то преимущество, что вы получаете все обновления, не скачивая всё по новой, с помощью простого *git pull*. У проекта будет *git*-адрес, например, <https://github.com/zfsonlinux/zfs.git>. Скачайте его командой *git* (установите пакет *git*, если команда недоступна):

```
git clone https://github.com/zfsonlinux/zfs.git
```

У вас создастся директория; войдите в нее по *cd* и прочитайте инструкции по компиляции. Репозитории для *git* и ей подобных обычно не имеют скрипта настройки, так что, как правило, сначала нужно запустить нечто вроде *autoreconf* для компиляции. *Readme* объяснит все необходимые действия.

Найдя исходник — неважно, каким образом — следуйте инструкциям по его компиляции и установке. Вам понадобится установить *autotools* и компилятор; в большинстве дистрибутивов имеется метапакет, который именуется наподобие *build-essential* и устанавливает всё необходимое для компиляции из исходника. Первый шаг этого процесса проверит, присутствуют ли в вашей системе зависимости, нужные для компиляции программ. Если в результате вы получите сообщение, что некая библиотека не найдена, перейдите в свой менеджер пакетов и установите эту библиотеку, а затем повторите процесс.

В этом процессе есть один постоянный прикол, когда вы получаете сообщение, что *libfoo* не установлена, хотя на самом деле она установлена. Причина в том, что дистрибутивы делят библиотеки на два пакета: первый содержит сами библиотеки, а отдельный пакет содержит файлы заголовков для библиотек. В обычном использовании эти файлы заголовков не нужны; библиотеке они не нужны, зато нужны компилятору для компиляции программ, связанных с этой библиотекой. Как правило, эти пакеты называются *libfoo-dev* в системах на базе Ubuntu/Debian и *libfoo-devel* в пакетах RPM. Установите пакет с заголовками, и ошибка исчезнет. Вы можете скачивать, распаковывать и компилировать программы как обычный пользователь, однако их установка требует привилегий *root* для копирования файлов в системные директории, поэтому финальный шаг будет примерно таким:

```
sudo make install
```

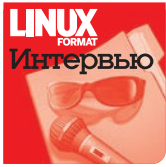
Смелее, попробуйте эти новые системы. Даже если вы решите, что для вас они пока не созрели, всё равно вы обязательно научитесь чему-то новому в процессе, а это никогда не помешает. **LXF**

» Многие проекты выкладывают свой самый свежий код на GitHub или подобный ему сервис; вам придется установить *git*, чтобы познакомиться с самыми последними версиями.

Джеймс Тернбалл

ДОКИНГ ЗАВЕРШЕН

Мэтт Хэнсон беседует с автором и активистом Docker о писательстве, развитии Linux и среде разработки Docker.



Джеймс Тернбалл [James Turnbull] — создатель бесплатных и открытых программ, разработчик и консультант в Docker Inc.

А заодно — технический директор в Kickstarter. Австралиец по происхождению, он сейчас живёт в Бруклине, штат Нью-Йорк, и является автором целого ряда книг, в том числе *The Docker Book* [Книга Докера], *Pulling Strings with Puppet* [Управлять «марионеткой»] и *The Logstash Book* [Книга о logstash].

Linux Format: У вас уже несколько книг...

Джеймс Тернбалл: Да, семь, на данный момент.

LXF: И все они технические?

ДжТ: Да — я бы и рад стать Джоном Гришмом [John Grisham, известный американский писатель], но в художественной литературе пока не нашёл своей темы... такой, чтобы принесла мне многотысячные тиражи и контракт с кинокомпанией! Так что — да, пишу о технике: о чём-то, что раньше было узкоспециальным, или о тех интересностях, за которые раньше никто всерьёз не брался.

Уверен, что с точки зрения издателей [это выглядит так]: техническая литература продаётся с трудом, и если вы не сможете гарантировать определённый уровень продаж, предложений [издательских] вы не получите. Поэтому я стараюсь писать о тех вещах, о которых вряд ли напишет кто-то другой: совсем новых или едва зародившихся, как когда-то было с *Puppet*. У них не было ни огромного сообщества, ни экосистемы; то же касалось и *logstash*, а теперь и Docker.

LXF: То есть книга про Docker у вас последняя.

ДжТ: Да, верно.

LXF: Вы отделяете писательство от основной работы. Можно ли сказать, что это в какой-то степени ваша страсть?

ДжТ: Так и есть. Меня очень напрягает, особенно в открытых сообществах, то, что документация всегда отстаёт от реальности. А нередко в проектах есть куча инструкций, тогда как реальная документация состоит из 20 постов разной содержательности и детальности, неизвестно где, и исходного кода. Если вы инженер и создаёте новую технологию, но при этом не в состоянии уделить достаточно времени её описанию — разве что кто-то сделает это за вас — вам придётся туго.

Мне представляется, что я пишу для того, чтобы люди с нулевыми знаниями по данной теме стали более-менее уверенными пользователями — не обязательно экспертами, но, по крайней мере, освоились в ней по окончании чтения. Поэтому мои книги — это 200 страниц чистой практической информации, которая поможет вам с нуля подняться до 60–70%. Расти ли дальше — уже будет зависеть от вас.

LXF: Да и диаграммы, и цифры, которых нет на tap-страницах, всегда оказываются полезны.



ДжТ: Именно, и мне кажется, несмотря на то, что тема документации в Open Source из года в год обсуждается на разного рода конференциях, ситуация с этим далеко не идеальная. Не во всех проектах распространение документации является приоритетом.

LXF: Как вы «познакомились» с Linux?

ДжТ: Попробую вспомнить... Пожалуй, где-то в середине 1990-х, или ближе к концу. Я был в восторге от Интернета, а один мой друг был одним из первых разработчиков ядра Linux. У него был дискетод, к которому требовался драйвер. И как-то он мне сказал: «Ты должен попробовать этот Linux!» В те времена дистрибутивы Linux были на дискетах, и пришлось загрузить кучу 3,5-дюймовок. Я тогда подумал: «Так вот на чём работает Интернет!»

Я тогда работал на чём-то вроде OS/400, и мы только начинали разбираться с TCP/IP. Все машины были на базе SNA (Systems Network Architecture), и тут меня осенило: «Ведь в Linux есть стек TCP/IP, а значит, я могу запустить web-сервер; да господи, есть же исходный код — если я что-нибудь сломаю, то смогу и починить!» И идея с сообществом мне показалась замечательной. Я подумал: надо же, это не только бесплатно, но ещё и даёт пространство для развития множества технологий. Это долгий процесс, но зато за ним интересно наблюдать. За все эти годы я ни разу ни от кого не слышал «Не нравится мне этот Linux». Даже крупные предприятия отзываются о нём в духе: «Open Source? Да, это круто», тогда как ещё лет пять назад говорили: «А-аа, Open Source, и чем же эта коварная лицензия нам грозит?» Приятно видеть такой прогресс. К тому же, с учётом повсеместной распространённости Linux, не удивлюсь,

если он выполняет до 30% всей рабочей нагрузки. Мне кажется, атавизмы вроде Unix и Solaris вскоре уйдут в прошлое, и Linux будет царить среди операционных систем, или, по крайней мере, встроённых систем на базе ядра. Думаю, это будет потрясающе!

LXF: Безусловно. Мы сами на это надеемся!

Сегодня мы видим, что многие из тех, кого слово "Linux" до сих пор пугает, ведь это не Mac OS и не Microsoft, вовсе используют «умные устройства», в основе которых — либо Linux, либо его производные, например, Android.

ДжТ: Если задуматься, этот выбор напрашивается сам. Создавая что-то на базе технологии встро-

О СТРАСТИ К ПИСАТЕЛЬСТВУ

«Документация в Open Source ещё далека от совершенства.»

енных систем, вы не возьмёте проприетарную ОС. Ведь это попросту невыгодно. Будь то холодильник, умеющий сам составлять список покупок, или какой-то бытовой прибор, создавать его вы будете на Linux. Здесь и выбора-то как такового нет, поскольку нет альтернатив.

LXF: Открывает ли подобное расширение сферы использования Linux какие-то перспективы для Docker?

ДжТ: Думаю, да. По-моему, ядро Linux можно смело назвать средой выполнения номер один. А значит, в основе Docker лежит надёжная технология, функции которой мы используем для создания контейнеров Docker. Так что в конечном итоге то доверие, которое людям внушает Linux, »



переходит и к нам. Мы не зависим от дистрибутива или операционной системы. Проект Docker связан чисто с ядром — чтобы его запустить, понадобится версия не ниже 2.8. Других зависимостей у него нет, и это нам очень нравится.

LXF: Как вы пришли в Docker?

ДжТ: Вскоре после того, как он стал открытым, в районе марта 2013, я начал к нему присматриваться. Проект был ещё на самом старте, и один коллега мне сказал: «Эта штука никогда не вытеснит виртуальные машины. Видали мы уже и Solaris Zones, и логическое разбиение. LXC тоже известно давным-давно. Очередной холостой выстрел. Круче гипервизора всё равно ничего нет». И я задумался над этим, вспомнил о местах, где раньше работал, о том, как громоздки эти виртуальные машины, как долго их собирать — а гипервизор ещё и создаёт лишнюю нагрузку. Задумался о том, на кого ориентирован Docker. Я понял, что Соломон [Хайкс, Solomon Hikes — основатель Docker] рассуждал так: «ОК, значит, самая обделённая вниманием группа пользователей у нас — разработчики приложений». И происходит это либо по недосмотру, либо намеренно, из боязни с ними связываться! Соломон поставил себе цель «создать технологический стек, в котором и виртуализация, и вычислительные ресурсы были бы вторичны по отношению к рабочему процессу. Суть последнего такова: „Я разработчик,

хочу писать код на ноутбуке, делать его портируемым и как можно скорее переносить с ноутбука на рабочий сервер, за что получу деньги, или тот бизнес, где он будет применяться, принесёт деньги».

Для большинства организаций это нетривиальная задача. Кому-то на неё понадобится полгода. Docker предлагает как бы «облегчить задачу портирования с помощью контейнера и вычислительного компонента и сделать это более эффективно», чтобы, тестируя своё приложение на своём хосте Docker в ноутбуке, я был уверен, что образ среды разработки является достоверным. И если мой код работает на ноутбуке, велика вероятность, что он заработает и на сервере. Я терпеть не могу клише, но это идеальный инструмент DevOps [методология, нацеленная на взаимодействие и интеграция разработчиков ПО и ИТ-специалистов, — прим. пер.] И с этим я глубоко солидарен.

Я сам вышел из Puppets Labs и много работал с сисадминами, а также посещал мероприятия Docker, где 60–70% аудитории составляют разработчики. Инфраструктура их вообще не интересует, для них важно одно: «эта штука облегчит мне жизнь, позволит внедрить мой код в систему, и операционисты уже не будут названивать мне в 3 часа ночи, жалуясь, что он не работает». И это очень важно.

LXF: Да уж, в таких звонках мало приятного.

Стало быть, цель Docker во многом состоит в том, чтобы выполнять работу стало проще и быстрее. Вы называете это «Docker'изацией». Не могли бы вы подробнее рассказать об этом процессе?

ДжТ: Конечно. По сути своей, Docker — это виртуализованная ОС, то есть контейнеры надстраиваются непосредственно на ОС, без всякого гипервизора. В основе Docker лежат два главных понятия: фаза создания и фаза выполнения. На первой происходит создание образов Docker. Для этого необходим так называемый Docker-файл. Существует куча инструкций по «созданию образа». Вначале берется базовый образ, например, Ubuntu или RedHat, и поехали: «на базовый образ Ubuntu/RedHat я устанавливаю Apache, PHP, или там WordPress, и добавляю свой исходный код. Сообщаю основные параметры настройки сети и указываю, что при запуске контейнера на базе моего образа должна выполняться такая-то команда». При этом не важно, кто я: разработчик приложений или сисадмин. Я просто создаю образ. Docker прекрасно подходит для совместного использования, так что готовый образ я могу загрузить на сервер Docker или в реестр Docker, и поделиться им со всей своей командой.

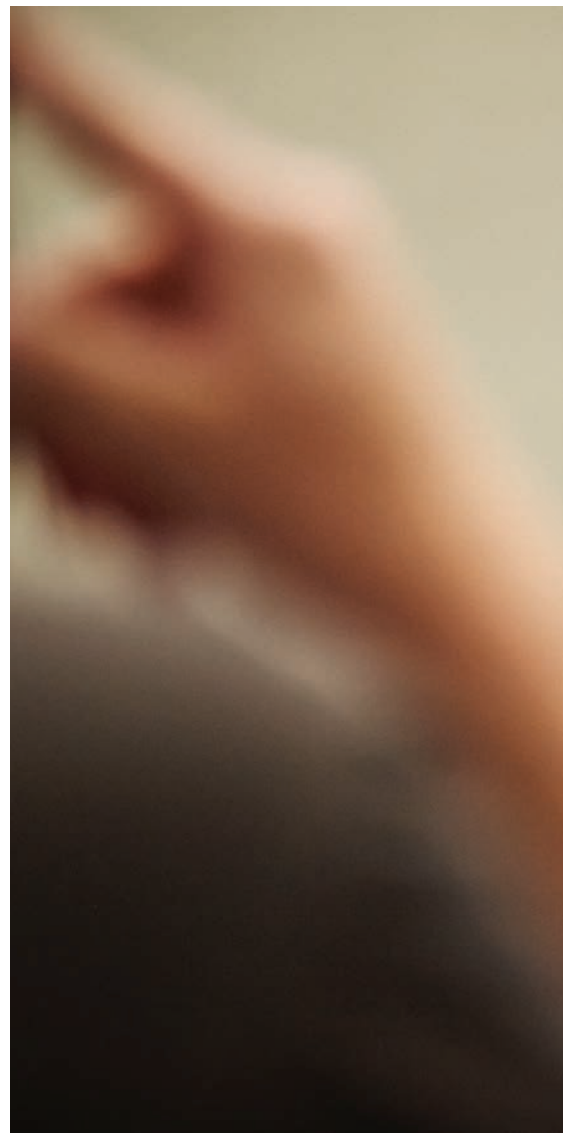
И тут кто-то скажет: «А добавлю-ка я туда новый образ WordPress». Потом кто-нибудь из разработчиков решит: «Создам-ка я новый контейнер, надо запустить Docker», и это уже будет стадия выполнения: «заодно создам новый образ. Ведь у всей моей команды и у всех, кто со мной работает, он будет абсолютно идентичен».

Затем я могу протестировать свой плагин WordPress или новую версию HTML, и если мне

захочется внести какие-либо изменения в образ, я запросто его обновлю. Образы создаются по принципу копирования при записи [copy-on-write], поэтому являются многоуровневыми. Так что при каждом изменении образа он будет отличаться от предыдущего. И речь не о сотнях и сотнях мегабайт, как при перестройке виртуальной машины. Это просто модификация кода, сколько бы там ни было, измеряться это будет в килобайтах. То есть ничтожно мало. Когда моё приложение готово, я просто сообщаю: «Вот последняя версия образа, со всеми обновлениями». Вновь отправляю его на сервер Docker и могу плодить на его основе контейнеры хоть из рабочей среды, хоть из среды тестирования. Всё очень легко и просто.

LXF: А если вы разместили образ, а затем самостоятельно внесли в него изменения и обновили его, остальные ваши коллеги об этом узнают?

ДжТ: У нас есть сайт, который мы называем Docker Hub, в знак почтения к GitHub, с помощью которого можно, в том числе, делиться образами. Он также будет уведомлять всех о создании



нового образа. Можно установить автоматическую привязку к файлу Docker, с которого создаётся образ. Разместить этот файл в репозитории рядом с кодом вашего приложения, чтобы при каждом обновлении образа он отправлялся на Docker Hub и распознавался там как новый файл Docker, а сам образ автоматически обновлялся. А можно сделать триггерное уведомление — работать с ним очень легко. Когда кто-то из членов команды объявляет о появлении нового образа и остальные начинают скачивать его с Docker, они получают только изменённую часть старого образа, а не весь целиком, сколько бы мегабайт он ни весил. Как правило, образы Docker небольшие. Это точно не гигабайты и не целая операционная система — только само ваше приложение и ещё кое-какая мелочёвка.

О ДОСТОИНСТВАХ DOCKER

«По исследованию IBM, Docker в 26 раз эффективнее виртуальной машины.»

LXF: Популярность Docker Hub воодушевляет?

ДжТ: Ну да. Думаю, сказалося то, что мы были свидетелями успеха GitHub в привлечении разработчиков и в качестве репозитория, и решили: «Раз мы хотим, чтобы люди делились своими образами Docker, именно такое нам нужно». И было очень приятно наблюдать стремительный рост числа подписчиков и посещений. На самом деле, в поисках какого-нибудь образа я частенько натываюсь на то, что кто-то уже создал образы для стеков. То, что в Oracle, к примеру, подумывают о создании образа WebLogic — уже здорово. Каждый раз, когда в команде заходит об этом речь, я думаю: «Ух ты, Oracle жаждет стать клиентом нашей компании со штатом в 45 сотрудников». Потрясающе.

LXF: Как появилась компания Docker и в чём заключается её миссия?

ДжТ: Прежде чем Docker превратился в Docker Inc., мы были компанией dotCloud — это продукт PaaS [платформа

как сервис], и Соломон Хайкс, тогда генеральный директор dotCloud, мечтал облегчить разработчикам задачу внедрения кода, сделав так, чтобы их приложения просто работали. Соломон посмотрелся к PaaS и подумал: «Это не совсем то... И сделать непросто, и абстрагировать, и применять». Сейчас облачные сервисы стали гораздо совершеннее, но работа с PaaS по-прежнему требует отдельного обслуживания и конкретных вариантов применения.

Благодаря нашему опыту работы с PaaS, мы очень много узнали о способах их применения разработчиками: как они разворачивают PaaS и запускают с их помощью свои приложения. Соломон возьми и скажи: «А не создать ли для всего этого абстракцию, образ и модель контейнера? Чтобы это был уже не просто PaaS, но и средство портирования с любых устройств, будь то ноутбук или штатный центр обработки данных, на какой-нибудь AS, вроде Easy2, или PaaS типа Cloud Foundry, или какой-нибудь внутренний сервер. Добавив вычислительный элемент и наладив его работу, мы сможем это сделать. И решим эту проблему». Суть проблемы была именно в портировании: «Код у меня есть, но в моей среде трудно заставить его работать, а перенести в другую среду ещё труднее». Нашей целью было максимально сгладить переход, облегчить притирку при переносе кода с места на место.

LXF: В чём главное отличие Docker от других средств виртуализации с гипервизором?

ДжТ: В его легковесной природе. На простейшем сервере с платформой виртуализации на гипервизоре один лишь гипервизор потребляет 10–20% ресурсов сервера. В случае с Docker это всего 1%. В недавнем исследовании IBM сравнивалась производительность Docker и виртуальной машины. И Docker оказался эффективнее в 26 раз. Так что если представить, во сколько вам обходятся несколько серверов, при том, что вы теряете по 10–20% от каждого и можете эти потери сократить, то это довольно выгодное предложение.

Создать образ Docker — дело минутное, чего не скажешь о виртуальной машине. Контейнеры Docker запускаются за доли секунды — виртуальных машин, которые загружались бы с такой скоростью, нет вообще. Даже среди тех, кто используют моментальные снимки. А здесь легко создать множество контейнеров, и так же легко от них избавиться. Они занимают гораздо меньше места, чем виртуальные машины. Это необычайно выгодное решение.

LXF: А чем принцип действия Docker отличается от традиционной виртуализации с гипервизором? Как LXC удаётся уменьшить размер контейнеров?

ДжТ: Основное отличие виртуализации с гипервизором от виртуальной ОС, что и представляют собой контейнеры Docker, таково: в первом случае у нас есть некая аппаратная часть, некая операционная система и гипервизор, и есть виртуальная машина, в которой своя гостевая





операционная система, а поверх неё — ваши приложения. Так что между вами и вашим оборудованием — куча промежуточных звеньев. В случае виртуальной ОС, есть демон Docker и контейнер. Промежуточных звеньев гораздо меньше, благодаря чему вы получаете более высокую производительность.

О БЕЗОПАСНОСТИ

«Мы открыто говорим, что можно или нельзя с Docker и насколько он безопасен.»

Контейнер Docker взаимодействует непосредственно с ядром операционной системы. То есть минуя какой-либо гипервизор. Приложению не нужно проходить через гостевую операционную систему, затем через гипервизор, и только потом в аппаратную часть. Сразу из контейнера, через демон Docker оно попадает в домашнюю ОС, а затем в аппаратную часть. Путь становится гораздо короче. В результате значительно упрощается процесс отладки — легко отследить, что происходит.

LXF: Безусловно, у этого решения есть свои преимущества, но с гипервизором виртуализация

всё же безопаснее. Стоит ли здесь идти на компромиссы?

ДжТ: Думаю, да. Всё дело же в риске, я правильно понимаю? Да, контейнер «уязвимее» виртуальной машины, но в мире разработки всё дело во времени цикла. В том, насколько быстро выполняется работа. И да, мы вполне уверены, что найдётся

целый ряд клиентов Docker, которым будет важно именно это. Чем меньше времени пройдёт с тестирования в стадии разработки — допустим, они занимаются только развёртыванием приложений

и не используют контейнеры для производства — до самого производства, тем выгоднее это будет для них.

Кроме того, в этой среде вопросы и требования безопасности не столь насущны. Хотя, безусловно, мы открыто говорим о том, что можно или нельзя сделать с помощью контейнера Docker и насколько он безопасен. Мы всегда говорим людям, что важно соблюдать определённые правила: не запускать в контейнерах процессы от имени root. Запускать такие приложения только на отдельном хосте Docker, чтобы не было более уязвимых и менее уязвимых контейнеров, когда через один можно заразить или взломать другой.

Мы рекомендуем использовать такие вещи, как iSCSI Linux на хосте Docker, чтобы, так сказать, укрепить контейнеры. Думаю, эти меры безопасности всегда стоит соблюдать. Но мы также говорим и о том, что в чём-то риск будет выше, чем при использовании виртуальной машины. И вашей организации предстоит решить, готовы ли вы пойти на этот риск, доверяя определённую работу контейнеру.

LXF: Полагаю, если открыто говорить об этих рисках, ваши клиенты смогут принять взвешенное решение.

ДжТ: Верно, и не в укор некоторым более традиционным инструментам виртуализации будет сказано, гипервизор — это по сути чёрный ящик. У платформ виртуализации уже был выявлен целый ряд потенциальных проблем и уязвимостей, которые, в особенности в случае с проприетарным ПО, совершенно не очевидны конечному пользователю.

Решение проблем безопасности, а также производительности и функциональности виртуализации — задача чрезвычайно трудная. Поэтому мы стараемся действовать от противного, как и поступает большинство открытых проектов: вот вам исходный код, вот обнаруженные ошибки, вот компоненты — всё выкладывается перед вами, а вам остаётся только принять обоснованное решение. **LXF**

softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



корпоративные
мобильные решения
на платформе Microsoft

SaaS

CLOUD

PROJECTS

гибридные облака

SOLUTIONS

remote workplace

SAM

поставка аппаратных
решений
юридическая
поддержка

education

Mirapolis

SECURITY
учебный
центр

разработка IT-стратегий

hardware
ERP&CRM
CAD
CAM
CAE

VIRTUALIZATION

web-разработка legal support

защита
персональных данных

SOFTWARE

публичные облака

САПР и ГИС

LINUX

программное
обеспечение

BI & BigData

SERVICES

WinSpace

информационная
безопасность

решения Microsoft

data leak
prevention

PaaS

разработка информационных систем
корпоративный портал
DeskWork

open
source

техническая поддержка

search

flexible

CONSULTING

PLATFORM

IaaS

частные
облака

protection

hosting

technical support

16+

8 (800) 100-00-23

+7 (495) 232-00-23

www.softline.ru

info@softline.ru



ЧУДНЫЙ Г-Н ФАЙРФОКС



Firefox отпраздновал десятилетие и превысил 80 миллионов загрузок для Android. И вот мы встретились с вице-президентом Mozilla, ответственным за *Firefox*.



Ноябрь 2014 г. ознаменовался десятой годовщиной web-браузера *Firefox*. Мы поговорили с Джонатаном Найтингейлом [Johnathan Nightingale], вице-президентом Mozilla по проекту *Firefox*, чтобы узнать, что нового сейчас в *Firefox*, и послушать его мысли о будущем Сети и браузеров.

Linux Format: Что нового в *Firefox*?

Джонатан Найтингейл (ДжН): Позвольте мне начать с постоянной присказки Mozilla: мы стараемся решать действительные ваши проблемы и предлагаем вам осознанный выбор. Это началось с запуска *Firefox* десять лет назад и продолжается по сей день.

Мы всё время ищем простейшие вещи, которые вы можете рассказать нам о своих онлайн-предпочтениях. Затем мы думаем, как представить вам это в понятном виде, чтобы вы начали видеть, как ваша информация утекает в Сеть. Так мы сделали с опцией запрета отслеживания Do Not Track; теперь это поддерживают все браузеры. Так мы сделали с WebGL, теперь это поддерживают все браузеры.

Ту же работу вы видите над кнопкой Забыть [Forget], это одно из нововведений в специальном юбилейном релизе *Firefox*. Она основана на простой идее. Если вы оставили следы в виде ссылок и понимаете, что вы не там, где хотели быть, или нажали на ссылку, или выполняете поиск, более уместный в Режиме приватности, но вы забыли его включить, то вы можете просто стереть этот факт. Вы можете попросить *Firefox* забыть последние пять минут, последние два часа или

последние 24 часа. Одним щелчком вы удаляете вашу недавнюю историю и cookie, закрываете все вкладки и окна и открываете новое окно, чтобы начать заново.

Ещё мы добавили в качестве одного из доступных способов поиска DuckDuckGo, поскольку поиск без отслеживания — это интересное предложение. Мы понимаем, что даже пользователи, не погружённые в техническую среду, беспокоятся о том, что их информация отслеживается и продаётся, и им не нравится, что это делают без их согласия.

Также мы вводим Помощника по Приватности [Privacy Coach] — дополнение для *Firefox* на Android, которое даёт наглядную информацию о возможностях *Firefox*, связанных с приватностью, таких как Do Not Track, Режим приватности, Режим гостя [Guest Browsing], блокировка cookie и удаление истории. И мы рады объявить о выпуске *Firefox Developer Edition*, созданном для

ДжН: Нынешний! Сегодня — замечательное время, ведь речь идёт о технических инновациях во всей Mozilla: новых функциях для пользователей, беспокоящихся о приватности; сотрудничестве с другими организациями, работающими над улучшением Сети; всех инструментах разработчика в одном особом браузере. Конечно, здорово иметь возможность оглянуться на 10 лет назад и увидеть, какой большой путь мы проделали, но ещё больше вдохновляет та энергия, с которой мы ведём ныне актуальные бои и делаем вещи, которые должны делать. Лучшего времени ещё никогда не было.

LXF: Как и зачем вы создали *Firefox Developer Edition*?

ДжН: Интернет как таковой — открытое пространство для взаимодействия, и это здорово, но иногда приводит к путанице. Есть немало проприетарных платформ разработки, которые не ла-

О ЗАЩИТЕ ПРИВАТНОСТИ

«Понять, как вас отслеживают, не должно быть так сложно, как сейчас. Это — ваша жизнь.»

разработчиков и тех, кто интересуется, из чего только сделана Сеть. Теперь у них есть место, которое можно называть домом, и не надо призывать следовать их привычкам браузер, ориентированный на потребление.

Мы также работаем (в партнёрстве с другими участниками) над частной инициативой, цель которой — улучшение защиты приватности в Сети. Мы будем проводить исследования и эксперименты и оттачивать технологии этой защиты.

LXF: Какой ваш любимый момент за прошедшие 10 лет *Firefox*?

дят друг с другом, и разработчикам приходится переключаться между разными инструментами, платформами и браузерами, из-за чего разработка более медленна и менее продуктивна. С проприетарными платформами легко создавать приложения для определённых магазинов, и это, очевидно, заманчиво, но разработчикам хочется создавать красивые вещи, которые сможет использовать каждый, на каждой платформе.

Firefox Developer Edition — первый браузер, созданный специально для разработчиков. Это стабильный *Firefox*, включающий все привычные инструменты разработчика и ряд новых

возможностей для упрощения web-разработки. Создателям web-приложений не придётся загружать добавочные плагины или приложения для отладки мобильных устройств.

LXF: Почему для вас важна приватность?

ДжН: Утечка данных неочевидна. Некоторые компании хотят получать выгоду от вашего непонимания происходящего. У вас есть право изменить ситуацию. Вы должны быть строже к нам и к другим.

Это не то же самое, что не заглядывать под капот вашей машины. Не заботиться о приватности уже нельзя. Мы стараемся помочь вам, предоставляя способы понять, кто что знает о вас и как вас отслеживают онлайн. Это не должно быть так жёстко, как сейчас. Это — ваша жизнь, и вы должны защищать её в Сети — так же, как вы защищаете её в физическом мире.

В вашей офлайн-жизни происходит много взаимодействий, на которые вы сами вольны давать или не давать согласие. Когда вы идёте мимо полки с кукурузными хлопьями, вы видите, что маркетологи приложили все усилия, чтобы привлечь ваше внимание, но никто не оспаривает факт вашего выбора покупки. Когда вы в Сети, вы не выбираете, кто увидит 10 ваших последних приобретений. Это странно и ненормально, мы должны это исправить.

LXF: Как вы думаете, какой будет Сеть спустя десятилетие?

ДжН: Есть четыре основных тренда, которые я вижу на ближайшие десять лет...

1. Весь мир окончательно осядет онлайн. Половина мирового населения уже в Сети, а остальные пользователи там скоро будут. Темпы только усилятся. Наша Firefox OS и другие инициативы упрощают доступ к Интернету без обладания смартфоном за £600. Десять лет назад Интернет был предметом роскоши. Через 10 лет мы будем думать об Интернете воистину глобальном, в таких масштабах, которые были невозможны 10 лет назад, при рождении *Firefox*.

Этот сдвиг будет большим и качественным. Те, кто присоединяется к Сети сейчас, будут иметь другие потребности по сравнению с теми, кто уже здесь. Это потребует от программ изменений и даст возможность появиться продуктам и функциям, отвечающим новым запросам. Мы рады это видеть — это одна из важных причин, почему мы вкладываемся в *Firefox* на всех платформах: мы глубоко убеждены в ценности всеобщего подключения к Сети.

2. Всё бóльшая часть жизни будет проходить онлайн. Когда *Firefox* появился, у меня было два подключения к Сети: одно дома и одно на работе. Сейчас у меня десятки вещей, связанных с Интернетом — телефоны и планшеты с Wi-Fi, Kindle с 3G, несколько компьютеров, и это ещё не всё. У меня нет термостата, холодильника или пылесоса с интернет-подключением, но люди уже такими обзавелись.

Через десять лет подключено будет всё, и возникнет вопрос: «Как нам организовать свою цифровую жизнь, как нам вести себя посреди этой

Сети Вещей, управлять ею, понимать её?» Перед Mozilla стоит насущная задача создать инструменты, с которыми пользователям будет проще управлять цифровой жизнью.

3. Данные станут сложнее. По мере появления новых возможностей подключения и возрастания числа устройств в Сети усложнятся и способы обработки информации. По-прежнему будет напряжённость в вопросе, мы ли контролируем данные или данные контролируют нас.

В Mozilla мы считаем, что контролировать должны вы, и собираемся продолжать делать инструменты, с которыми утечки данных будут заметнее и которые дадут вам возможность выключить их, или дать разрешение на них, но обозначить, что вам это не нравится. Мы продолжим нашу политику пропаганды и продолжим строить технологическое подтверждение того, что создавать вещи, уважая людей, которые ими пользуются — возможно.

4. Интернет, каким мы его знаем, уйдёт в прошлое. Не в том смысле, в каком автомобили и холодильники заменяются другими, получше. Я имею в виду, что корпорации заинтересованы в том, чтобы их надзиратели знали всё о том, как вы выходите в Сеть и какие приложения для этого используете. У них красочные логотипы и приятные конференции про дизайн, но не стройте иллюзий: их бизнес-модель — стать посредником между вами и всем вокруг.

Эти надзиратели постараются сделать самый привлекательный, интегрированный набор продуктов, ведь это их деньги. Я не осуждаю их за желание зарабатывать, но не хочу, чтобы мою жизнь контролировали бизнес-модели компаний. И не считаю такое объяснение наивным. Это реальная угроза.

Дело в том, что это гораздо шире, чем обычное техническое обсуждение, чей набор технологий лучше. Не в том, кто в будущем победит — приложения или браузер; речь о гораздо большем. Вопрос в том, сохраним ли мы Сеть как глобальный рынок идей, где, если у вас есть идея и она дерзкая — она изменит мир. Если пользователи не победят, это станет проблемой.

LXF: Как будет выглядеть браузер будущего?

ДжН: Люди уже принимают как должное, что простейший браузер умеет отображать страницы,

взаимодействовать с другими браузерами и выдавать им то, что они искали, и это хорошо. Теперь все осваиваются в мире приложений, но я считаю, что нужны браузеры, предоставляющие пользователям поддержку на более глубоком уровне. Думайте о браузере не как об окне, через которое вам видно то, что за ним, а как о программе, которая работает ради вас. Браузеру ещё есть куда расти, чтобы превратиться в помощника или друга, которому вы доверяете в Сети, который даёт понятный вам выбор и предугадывает ваши потребности, и всё это без потери привычной производительности.

Между работой в Сети в 2014-м и в 2004-м — масса различий. И очень больших — здесь в нашем разговоре и появляются приложения; но ещё важнее взаимосвязанность устройств и то, что вы должны получать единый пользовательский опыт, перемещаясь между работой и домом или между компьютером и смартфоном.

LXF: Что сегодня делает Firefox особенным?

ДжН: По всем объективным характеристикам, которые я могу перечислить — хотя и признаю, что я не объективен — я сегодня считаю *Firefox* лучшим. Он быстрый, он красиво спроектирован, он хорошо настраивается. *Firefox* самый быстрый по всем показателям, но я смотрю на большее, чем просто показатели каждого приложения. Некоторые браузеры могут показывать удивительную скорость JavaScript или быть такими же настраиваемыми, но никто из них не сравнится с нами по независимости. Людям важно, что у нас есть миссия, есть ценности — ни один другой браузер не выполняет этот пункт.

Наши конкуренты — Google, Apple и Microsoft. Всё это громадные корпорации, чья бизнес-модель в том, чтобы заманить вас на свою платформу и привязать к ней. Мы вместо этого ведём вас в Сеть — большой, взаимосвязанный, всемирно доступный печатный станок нашего века, без надзирателей.

LXF: Какие функции вы хотите видеть в следующем Firefox?

ДжН: *Firefox* хорош тогда, когда он мне помогает. Десятилетие назад это означало дать мне вкладки, чтобы у меня не было сотен открытых окон одновременно, потому что ими сложно управлять.



В других браузерах тоже использовалась концепция вкладок, но *Firefox* популяризовал её, сделал мою онлайн-жизнь проще. 10 лет назад *Firefox* помог мне, блокируя всплывающие окна. Это было спорным моментом, так как считалось, что браузер должен быть объективен и не должен вмешиваться в то, что делает web-страница. Я не согласен с этим, я считаю, что у нас должна быть позиция. Тогда наша позиция была в том, что всплывающие окна раздражают, и мы сделали блокировщик, чтобы от них избавиться.

Взглянем на мои сегодняшние проблемы. Сегодня это переизбыток устройств. Я хочу видеть в *Firefox* функции для упрощения переключения с одного устройства на другое и их синхронизации, чтобы пользоваться ими как одним. Я бы хотел иметь возможность начать делать что-то на ПК и затем продолжить на телефоне или планшете с того же момента, где остановился.

Я обеспокоен насчёт приватности как в Сети в целом, так и на моих устройствах. Я хотел бы дать телефон другу в ресторане, чтобы он быстро посмотрел что-то, но не проник в мою личную информацию. Как мне это сделать? Я хотел бы поделиться ноутбуком с соседом в колледже, однако у меня есть секреты. Как мне это сделать?

LXF: Почему так важна настраиваемость?

ДжН: *Firefox* быстр, безопасен и приятен. Он относится к вещам, работающим из коробки. Для этого ничего не нужно делать. Но когда мы разгова-

и немножко меняю их в зависимости от того, чем я занят. Если я что-либо планирую, я перенастрою его и найду расширение, которое разместит мои вкладки сбоку, чтобы было проще ими управлять, или размещу их в окнах и примусь использовать группы вкладок. Когда я соберусь работать над чем-нибудь другим, я снова всё перенастрою.

Я из тех людей, что накапливают закладки — у меня их сотни, поэтому при каждом удобном случае я навожу в них порядок.

LXF: Какое ваше любимое расширение для Firefox?

ДжН: Меня восхищают вещи, которые дают людям лучший обзор за происходящим с ними в Сети. Люди умны и хотят понимать, как работает Интернет. Он сложен, и мы должны сделать понимание и управление проще, поэтому я использую расширения вроде *Lightbeam*, разработанного в Mozilla. Оно показывает, как посещаемые мною сайты наблюдают за моим поведением и исполь-



об онлайн-играх, я просто восхищаюсь современными возможностями.

Игры были одним из последних оплотов ПО, которое надо скачивать — игра работала, только если установить её на компьютер или консоль и запустить локально. В октябре Mozilla выпустила пакет из восьми онлайн-игр вместе с Humble Bundle. Он показывает, что у вас есть возможность играть в масштабные, полноценные, интерактивные игры в Сети без плагинов, без загрузки и установки чего-либо.

Это становится возможным благодаря работе, которую мы ведём, чтобы сделать JavaScript настоящим быстрым и упростить разработчикам перенос существующего кода в JavaScript и Сеть. Это связано и с нашей работой в области графического ускорения с WebGL, с которым можно делать красивые вещи для Интернета.

Мы сделали Сеть более открытой, чтобы разработчики игр могли извлечь те её преимущества, которые другие виды приложений извлекают уже 10 лет. Поэтому мы видим, как основные игровые движки приглядываются к Интернету в качестве игровой платформы. Сеть — это потрясающее место для разработчиков, потому что распространение в ней максимально простое и вам не нужно уговаривать людей что-то установить. Разработчики игр начали осознавать, что пришло время сказать: «Я хочу, чтобы игра стала доступна на PC, на Xbox и в Сети».

LXF: Насколько для вас была важна работа сообщества волонтеров за эти 10 лет?

ДжН: Мировое сообщество волонтеров — это необходимая жизненная сила для нашего процесса разработки. Над *Firefox 33* работали около 500 человек, и более половины из них были волонтерами.

К тому же благодаря работе сотен волонтеров по локализации *Firefox* доступен на 90 языках. Это учит скромности, и это невероятно ценный опыт — работать вместе с людьми, которые не получают денег, которые тратят своё время, наблюдая за всем, что мы делаем для *Firefox*, собирают все тексты и работают с ними, наблюдая за всем, что мы пытаемся выразить, и перенося эти идеи на свой язык. Сообщество волонтеров — это сердце того, что у нас есть. **LXF**

О ВАЖНОСТИ НАСТРАИВАЕМОСТИ

«Кнопка „Настроить“ — единственный элемент меню, который нельзя убрать. Наша ключевая ценность.»

риваем с пользователями, они говорят нам, что ценят нашу индивидуальность и независимость. И мы ценим индивидуальность наших пользователей в ответ.

Итак, настраиваемость важна, потому что мы хотим, чтобы вам было как можно проще получить браузер, работающий так, как вам надо, хотите ли вы изменить его установкой темы или глубокой перенастройкой функциональности с помощью ряда расширений и изменения параметров.

С браузером вы проводите больше времени своей жизни, чем с любым другим приложением или интерфейсом. Поэтому — да, для нас важно, чтобы вы могли взять и изменить его. Кнопка Настроить — это единственный элемент меню, который нельзя убрать из вашего браузера, потому что мы верим: вы всегда, постоянно должны иметь возможность изменить браузер. Это наша ключевая ценность.

LXF: Как настроили Firefox лично вы?

ДжН: Мой браузер живёт тем же, чем я — и это заметно! Я тщательно настроил его с помощью расширений и моих персональных параметров

зуют мои данные, и проливает свет на то, какие сторонние сайты могут использовать данные с раннее посещённого сайта.

Мы не единственные, кто исследует вопрос, как сделать Сеть прозрачнее для вас, и меня это очень радует. Существуют другие расширения, например, *Privacy Badger* и *Ghostery*, основанные на тех же идеях. Их я всегда рекомендую, когда кто-то спрашивает меня об информации и приватности в Сети.

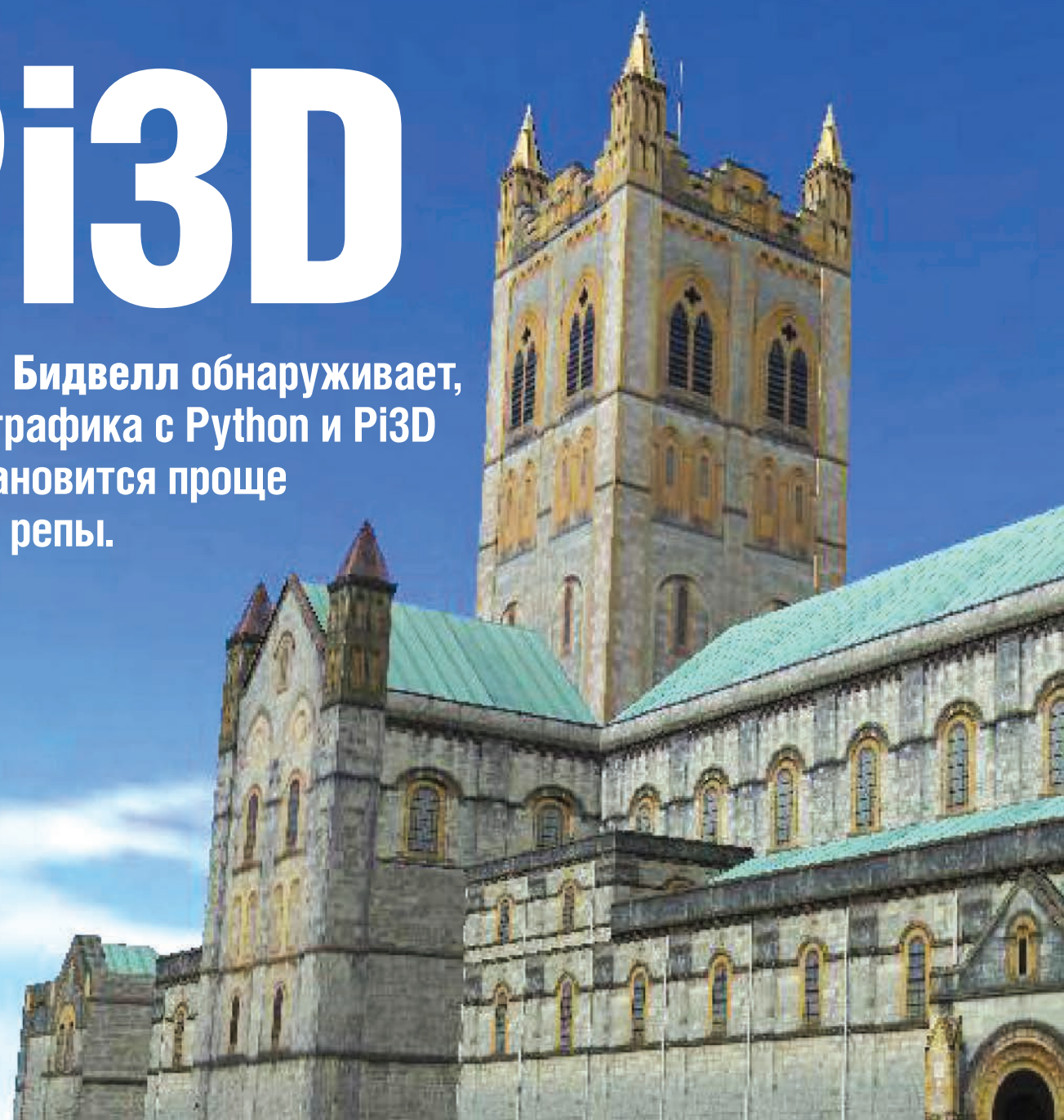
Всегда интересно видеть реакцию людей, когда они поработают с одним из этих расширений какое-то время — они начинают больше понимать о том, как их отслеживают. Это вызывает новые вопросы и всегда меняет взгляды пользователей на посещаемые сайты и применяемые этими сайтами инструменты для слежки.

LXF: Когда вы не работаете, используете ли вы Firefox для игр?

ДжН: Да. В разное время я использовал Интернет для игр разными способами. Игры, в которые я играю сегодня, лучше, чем те, в которые я играл пару лет назад, а игры, в которые я буду играть через год, будут сногшибательны. Когда я думаю

Pi3D

Джонни Бидвелл обнаруживает, что 3D-графика с Python и Pi3D на Pi становится проще пареной репы.



Несмотря на крошечный форм-фактор и относительно низкую мощность CPU, Pi может похвастаться графическим оборудованием на профессиональном уровне. В частности, Broadcom VideoCore IV — двухядерное устройство, поддерживающим OpenGL ES 2.0 и декодирование на уровне оборудования 1080p h.264 видео на 30 кадров в секунду. Всё это хорошо и мило, но что такое OpenGL ES и как мы можем использовать его, чтобы создать симпатичную графику? Хороший вопрос, дорогой читатель, и так уж вышло, что данная статья как раз посвящена ответу на него.

В OpenGL имеются привязки для многих языков, но их прямое использование не для слабых духом: программирование на OpenGL ничуть

не похоже на программирование на Python. К счастью для питонцев, есть несколько опций дорваться до всех прелестей 3D, не обременяя мозги всякими чуждыми концепциями. Например, мы рассматривали Intel *Clutter* (приложив библиотеку *Cogl*) в LXFI33. Это по-прежнему

«К счастью для питонцев, есть несколько опций дорваться до прелестей 3D.»

популярный способ работы: поверхности выделяются в Stages [Сцены], и то, что перемещается на Stage, становится Actor [Актёр]. *Clutter* может даже работать с X-сервером, что очень хорошо для легковесных демо-версий. Правда, чтобы использовать OpenGL ES, его нужно

подлатать и перекомпилировать, о чём вы можете прочитать на сайте Raspberry Pi (<http://bit.ly/ClutterandCoglonPi>).

Однако вам будет приятно узнать, что есть ещё более простой способ овладеть мощностью 3D на Pi, благодаря отличной работе Тима Скиллмена [Tim Skillman], Пэдди Гонта [Paddy Gaunt] и Тома Ричфорда [Tom Ritchford].

В 2011 году Скиллмен разместил экспериментальный код на форумах Raspberry Pi, и очень скоро проект, встреченный с большим воодушевлением и энтузиазмом, пошел в бурный рост. Официальный релиз читайте здесь: www.raspberrypi.org/pi3d. Но вам, очевидно, уже не терпится приступить к делу, так что следуйте инструкциям (см. *Установка Pi3D на Raspbian*, ниже), и всё вам будет.



➤ **Аббатство Бакфаст: помните, их «настройка» вовсе не предполагает наличия оздоровительных или медицинских свойств.**

Pi3D лучше всего изучать на практике, так что давайте начнём с файла **Earth.py** в директории демо. Он научит нас работать со сферами, текстурами, вводом с клавиатуры и немного с небесной геометрией.

Поскольку мир переходит — или должен переходить — на Python 3 [про него будет статья в **LXF195**], мы начнём с восходящей совместности, благодаря любезности модуля `__future__`. Затем мы импортируем некоторые тригонометрические функции, поскольку вы недалеко уйдёте без синусов с косинусами и самого модуля `pi3d`. Самое интересное начинается с настройки важнейшего объекта — `DISPLAY`. Поскольку от его существования зависит большинство других объектов, большинство проектов Pi3D с ходу начинают использовать эту строку. Объект поддерживает информацию о времени, что полезно для анимации. Вы можете передать размеры экрана в функцию `create()`, мы используем небольшое окно 50×50, где зададим чёрный матовый фон:

```
DISPLAY = pi3d.Display.create(x=50, y=50)
DISPLAY.set_background(0,0,0,1) # r,g,b,alpha
```

Теневые делишки

Частью волшебства OpenGL (и OpenGL ES) являются шейдеры [shade — *англ.* тень]. Шейдеры — это программы, написанные — сюрприз! — на языке `shader`, в нашем случае GLSL, и обрабатывающие отражения, тени и прочие взаимодействия между светом и поверхностями, объёмами и точками. GLSL использует C-подобный синтаксис и разработан ради задействования большого количества экземпляров шейдера на современном графическом оборудовании. Основная идея заключается в наличии множества небольших программ-шейдеров, работающих параллельно, вместе создавая сложные и удивительные результаты.

Pi3D мудро оставляет мрачные подробности своей реализации шейдера за кулисами. Но это не помешает нам добиться интересных эффектов для нашей Земли, лун и звёзд:

```
shader = pi3d.Shader("uv_light")
```

```
shinesh = pi3d.Shader("uv_reflect")
```

```
flatsh = pi3d.Shader("uv_flat")
```

Шейдер `uv_light shader` использует направления света и тени для создания 3D-эффекта, в отличие от `uv_flat`, который тупо выполняет рендеринг текстур без каких-либо цветовых трансформаций. Шейдер `uv_reflect` отражает одно изображение в другом. Мы будем использовать его для отражения рельефа на поверхностях Луны; в луне меньшего размера мы также отразим звёзды.

Демо-версии идут с самыми разными образами в поддиректории `textures`. Файлы PNG здесь содержат информацию о прозрачности, что очень удобно, потому что нам нужно будет знать, что происходит позади них: это, в конце концов, руководство по 3D. Например, мы скоро наложим на поверхность нашей Земли в основном прозрачный файл `earth_clouds.png`, чтобы создать на ней атмосферный воздух. Потому что сплошное солнце всё превращает в пустыню, как сказал один мудрец. Аргумент `True` в первой строке указывает, что для наших облаков нужна частичная прозрачность. Сначала загрузим все текстуры:

```
cloudimg = pi3d.Texture("textures/earth_clouds.png", True)
```

```
earthimg = pi3d.Texture("textures/world_map.jpg")
```

```
moonimg = pi3d.Texture("textures/moon.jpg")
```

```
starsimg = pi3d.Texture("textures/stars2.jpg")
```

```
watimg = pi3d.Texture("textures/water.jpg")
```

```
moonbmp = pi3d.Texture("textures/moon_nm.jpg")
```

Текстуры не имеют никакой ценности без поверхности, на которую их накладывают. Мы нарисуем Землю и Луну, представив их в виде сфер. Для Земли зададим две сферы: одна будет показывать подробности поверхности, а другая (с немного большим радиусом) обеспечит рендеринг атмосферы и прочих теневых эффектов.

У нас также будет две луны, что даст нам иерархическую систему вращения, в которой меньшая луна вращается вокруг большей, а та, в свою очередь, вращается вокруг Земли. Мы также зададим плоскость, на которой будем рисовать

Установка Pi3D на Raspbian

Pi3D имеет несколько зависимостей, включая некоторые заголовки, которые вы найдёте в репозиториях Raspbian. Так что сначала сделайте обновление, а потом установите нужные пакеты:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install python-dev pythonsetuptools
libjpeg-dev zlib1g-dev libpng12-dev libfreetype6-dev
```

Pi3D и библиотеки Pillow Python, от которой он зависит, пока нет в репозиториях Raspbian, но не бойтесь! Они есть в репозиториях *Pip*, так что надо установить *Pip* и использовать его, чтобы их найти:

```
$ sudo apt-get install pip
```

```
$ sudo pip install Pillow
```

```
$ sudo pip install pi3d
```

По умолчанию модели Pi с 512 МБ, В и В+, отводят 64 МБ на GPU. Хотя многие демо-версии Pi3D отлично поладят с подобным соотношением, некоторым из них потребуется чуть больше. Изменить распределение памяти GPU среди других системных настроек можно с помощью утилиты `raspi-config`:

```
$ sudo raspi-config
```

128 МБ будет вполне достаточно, они даже позволят вам запустить несколько приложений рабочего стола.

Демо-версии Pi3D можно клонировать из GitHub:

```
$ cd ~
$ git clone git://github.com/pi3d/pi3d_demos
```

У вас создастся директория `~/pi3d_demos`, из которой вы, например, сможете исследовать аббатство Бакфаст:

```
$ cd ~/pi3d_demos
$ python BuckfastAbbey.py
```

К нашему глубочайшему сожалению, фирменный магазин Аббатства, где продаётся знаменитая монастырская крепёжная настройка, там почему-то отсутствует.

звёздное небо фона. Помимо указания используемых радиусов и центров, структура Sphere также принимает дополнительные параметры, **slices** и **sides**, которые устанавливают количество широт и сегментов, по которым создаётся сфера. Увы, даже в наше время мы по-прежнему зависим от пикселей и прямых линий, так что нам не нарисовать настоящую сферу. Зато мы можем найти простой и красивый способ работы обычным вводом с клавиатуры — его мы применим позднее с единственной целью: завершить программу:

```
mysphere = pi3d.Sphere(radius=2, slices=24,
sides=24, name="earth", z=5.8)
mysphere2 = pi3d.Sphere(radius=2.05, slices=24,
sides=24, name="clouds", z=5.8)
mymoon = pi3d.Sphere(radius=0.4, slices=16,
sides=16, name="moon")
mymoon2 = pi3d.Sphere(radius=0.15, slices=16,
sides=16, name="moon2")
myplane = pi3d.Plane(w=50, h=50, name="stars",
z=30)
mykeys = pi3d.Keyboard()
```

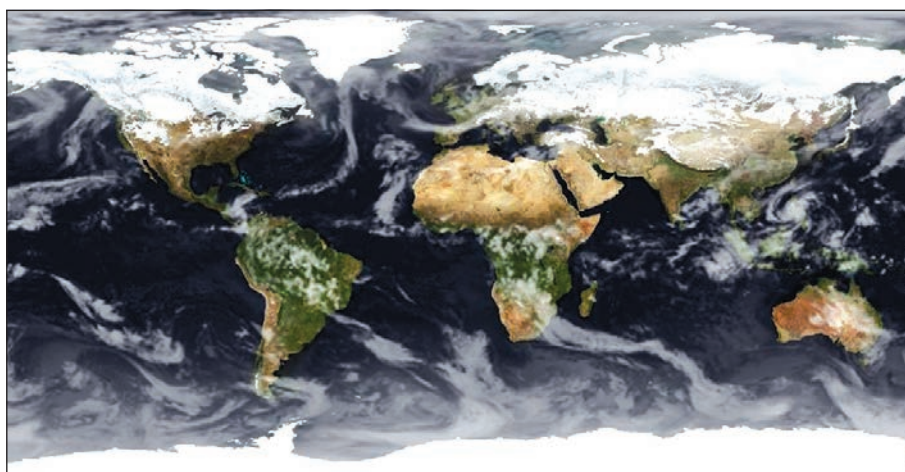
Если вы не укажете для вашей сферы координату x, у или z, она будет помещена в начало координат.

Небесный балет

Мы описываем орбиту Луны вокруг Земли и второй луны вокруг Луны с помощью параметров **rot1** и **rot2** соответственно. Они будут увеличиваться по мере прохождения основного цикла (к которому мы подбираемся). Радиус каждой орбиты задаётся **m1Rad** и **m2Rad**. Если установить орбиты соответственно на 90 и 0 градусов, это будет означать, что (с нашей точки зрения) Луна находится перед Землёй, а малая луна прилегает к большой по горизонтали.

```
rot1=90.0
rot2=0.0
m1Rad = 4
m2Rad = 0.6
```

Помимо вращения вокруг других тел, мы также вращаем Землю и обе луны вокруг своих осей у. Ось Y соответствует на вашем экране направлению вертикали, как в *Minecraft*, поэтому



► Нам нужны облака, чтобы сделать дождь, и альфа-каналы, чтобы через них проходил свет.

вращение вокруг этой оси подобно насаживанию Земли на ось, проходящую через полюсы и затем её раскручиванию, но не столь деструктивно. Мы вращаем облачную сферу со скоростью, немного превышающей скорость вращения Земли, что создаёт очень привлекательный эффект и намного точнее, поскольку облака не привязаны к Земле. И, глядя, как всё крутится, мы вращаем также фоновое звёздное небо, поскольку оно двумерно: единственная ось вращения для него — это ось Z.

Мы перерисовываем сферы **moon** и **moon2**, вычисляя их расположение с помощью обычной школьной тригонометрии и новых координат. Объект DISPLAY, который мы так удобно настроили, даёт нам основной цикл событий, и мы применим его для управления нашим небесным балетом:

```
while DISPLAY.loop_running():
myplane.rotateInCZ(0.01)
mysphere.rotateInCY(-0.1)
mysphere2.rotateInCY(-0.14)
mymoon.position(mysphere.x() +
m1Rad*sin(rot1), mysphere.y(), mysphere.z()
- m1Rad*cos(rot1))
mymoon.rotateInCY(-0.1)
mymoon2.position(mymoon.x() -
```

```
m2Rad*sin(rot2), mymoon.y(), mymoon.z() +
m2Rad*cos(rot2))
mymoon2.rotateInCZ(-0.61)
```

Теперь воспользуемся шейдерами для добавления к нашим небесным телам текстур и симпатичных эффектов. Шейдер отражения, использованный для лун, берёт парочку чисел для текстур, которые указывают количество элементов и, соответственно, силу отражения. Облака нужно рисовать последними, иначе эффект прозрачности не сработает: вы не можете впоследствии добавлять объекты, затемнённые полупрозрачностью, когда **blend = True**, так что безопаснее всего добавить подобные текстуры в самую последнюю очередь.

```
mysphere.draw(shader, [earthimg])
mymoon.draw(shinesh, [moonimg, moonbmp],
6.0, 0.0)
mymoon2.draw(shinesh, [watimg, moonbmp,
starsimg], 3.0, 0.8)
myplane.draw(flatsh, [starsimg])
mysphere2.draw(shader, [cloudimg])
```

Теперь мы увеличиваем параметры скорости вращения — меньшая луна вращается вокруг большей примерно в четыре раза быстрее, чем большая Луна вращается вокруг Солнца:

Что такое OpenGL

Без сомнения, вы слышали об OpenGL, унифицированном API для общения с графическим оборудованием. Этот язык появился в 1992 году в Silicon Graphics: там решили, что открытие кода стандарта будет отличной возможностью ослабить его конкурентов. Это неплохо сработало, но потом появился Microsoft Direct3D. Но тем не менее OpenGL выживает, и одной из причин является OpenGL ES. Это поднабор OpenGL, который используется на мобильных устройствах, встраиваемых системах и некоторых игровых консолях.

В отличие от привычных видеокарт настольных ПК, этим машинкам часто не хватает поддержки

множества регистров и встроенной в чип поддержки плавающей запятой, поэтому нужно кое-что соответствующим образом переработать. Но принцип остаётся тем же: получить универсальный метод для эффективной загрузки текстур, света и перспективных вычислений на графическое оборудование.

Подобно Open GL, OpenGL ES разработан, без лицензий и пошлин, Khronos Group, консорциумом промышленных гигантов и академических институтов. Помимо OpenGL, Khronos также координирует разработку OpenGL, WebGL, EGL и ещё нескольких видеостандартов.

В августе Khronos приняла в свои ряды нового члена: Microsoft. Однако было бы параноией считать, что это коварная попытка уничтожить OpenGL изнутри. Неоспоримый факт тот, что, на самом деле, сохранение стандартов для мобильной графики открытыми представляет общие интересы, и в общих интересах, чтобы все основные игроки поддерживали и вносили свой вклад в эти стандарты. Испытанная стратегия MS — поддержать, надстроить и уничтожить — здесь не сработает, поскольку с точки зрения других участников (например, таких, как AMD и Nvidia) это исключительно неуместно.

```
rot1 += 0.005
rot2 += 0.021
```

Поскольку ранее мы немало намучились с настройкой объекта `Keyboard`, было бы жалко его не употребить. Мы используем два клавиатурных события — нажатие на `Esc` (индекс клавиши — 27) завершит программу, а нажатие на `p` (индекс клавиши — 112) сделает экранный снимок.

```
k = mykeys.read()
if k > -1:
    if k == 112:
        pi3d.screenshot("earth1.jpg")
    elif k == 27:
        mykeys.close()
        DISPLAY.stop()
        break
```

Размытые линии/сферы

Что касается предоставленной демо-версии Земли, спокойно делайте с ней всё, что сочтёте нужным. В порядке альтернативы, можете продолжить вместе с нами. Начнём мы с размытия глубины резкости Луны, чтобы она выходила из фокуса, когда она к нам слишком близко или слишком далеко.

Чтобы проделать этот трюк, начнём с вызова модуля `Defocus`. Поместите следующую строку где-нибудь перед основным циклом — например, после строк, указывающих шейдеры:

```
defocus = pi3d.Defocus()
```

Расфокусировка работает посредством помещения стандартных вызовов `object draw()` внутри блока, ограниченного `start_blur()` и `end_blur()`. Объекты, «нарисованные» внутри этого блока, обчисляются в буфере и на экране не появляются. Чтобы сделать их видимыми, используйте метод `blur()`, который будет обчислять их с соответствующим для определённого расстояния размытием. Поэтому закончите строку, начинающуюся с `mymoon.draw`, следующим образом:

```
mymoon.draw(shinesh, [moonimg, moonbmp],
            6.0, 0.0)
defocus.end_blur()
```

Метод размытия, который и выполняет функцию отрисовки, берёт три дополнительных аргумента (помимо имени объекта `Shape` для рисования): фокусное расстояние; расстояние, дальше



› 3D-модели легко импортировать в формат .egg — архивный формат файлов Panda3D.



› Читатели со слабым зрением, не тревожьтесь: дело не в вас, просто Луна и правда не в фокусе.

(или ближе) которого всё будет максимально размыто; и степень максимального размытия. Установим нулевую плоскость на $z=0$, и поскольку орбита нашей Луны имеет радиус 4 единицы, установим второй параметр в 3. Задание максимального размытия чрезмерным приведёт к образованию полос, но экспериментально мы выяснили, что разумной настройкой здесь будет 5. Выполните это следующей строкой:

```
defocus.blur(mymoon, 0, 3, 5)
```

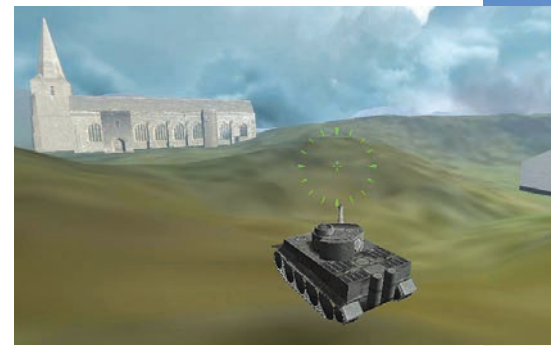
Человек с кинокамерой

Потеха на этом не заканчивается: добавив объект `Camera` к нашим действиям, мы можем полностью погрузиться в нашу систему из трех небесных тел. Используя чуть-чуть тригонометрии и наш уже реализованный объект `Keys`, можно переместить нашего небесного наблюдателя и изменить точку обзора. Нужно добавить функцию `radians` к тригонометрическим функциям, которые мы уже импортировали из модуля `math`. Теперь настроим объект `Camera` и инициализируем некоторые свойства для него после объявлений `DISPLAY`:

```
CAMERA = pi3d.Camera()
rot = 0
tilt = 0
rottilt = True
camRad = 5
```

Для запуска любых изменений позиции или ориентации камеры будем использовать булевскую переменную `rottilt`. Поворот и наклон камеры делается легко, однако изменение ее радиуса (определяемое `camRad`) требует стандартной жутки сферической тригонометрии, которой были посвящены упражнения в баллистике наших статей о `Minecraft` для `Pi` [см. Академия кодинга, стр. 86 LXF185 и стр. 84 LXF186]. Итак, началом основного цикла будет

```
while DISPLAY.loop_running():
    if rottilt:
        CAMERA.reset()
        CAMERA.rotate(tilt, -rot, 0)
        CAMERA.position(camRad * sin(radians(rot)) *
            cos(radians(tilt)), camRad * sin(radians(tilt)),
            -camRad * cos(radians(rot)) * cos(radians(tilt)))
```



› Ваши мечты поуправлять танком «Шерман» станут явью всего за пару строк кода.

```
rottilt = False
```

Теперь не помешает настроить кнопки для управления камерой Земли — мы употребим стандартные `W`, `A`, `S`, `D` для вращений и `+/-` для приближения и удаления.

Измените начало блока работы с кнопками следующим образом:

```
if k > -1:
    rottilt = True
    if k == 112:
        pi3d.screenshot("earth1.jpg")
    elif k == 119:
        tilt += 2.0
    elif k == 115:
        tilt -= 2.0
    elif k == 97:
        rot -= 2
    elif k == 100:
        rot += 2
    elif k == 61:
        camRad -= 0.5
    elif k == 45:
        camRad += 0.5
```

Отныне у вас есть и действия, и камера — так почему бы не призадуматься о добавлении света, скажем, какого-нибудь далекого солнца? Документацию вы найдете на <http://bit.ly/Pi3DDocs>, но предоставленные пакетом демо-версии отлично знакомят со всеми имеющимися структурами. LXF

LMDE Betsy:

Есть ли жизнь до релиза?

Алексей Федорчук проник на кухню разработчиков через черный ход и был встречен вполне приветливо.

Тема этой статьи не совсем обычна: она посвящается системе, релиза которой на момент сочинения этих строк ещё нет. Но которую, тем не менее, вот уже более двух недель можно не просто пощупать в тестовом режиме, как это обычно предлагается для всякого рода бета-версий и релиз-кандидатов. Нет, её уже сейчас можно полноценно применять на практике — без проблем на типовых задачах и с некоторыми, но достаточно простыми, ухищрениями — на задачах не совсем стандартных. Ибо речь здесь пойдёт о Linux Mint Debian Edition, релиз которой за номером 2 известен под партийной кличкой Betsy.

Предыстория Betsy

О дистрибутиве Linux Mint слышали, наверное, все, хоть каким-то боком прислонившиеся к миру Linux и FOSS. Ответившись от Ubuntu в ноябре 2006 года, он быстро превзошёл своего родителя если не по распространённости (ибо количественные данные на сей предмет для всех дистрибутивов условны и неопределённые), то уж точно по известности, причём не в очень узких кругах. Поскольку уже в первый год своей жизни вошёл в десятку сильнейших по рейтингу Distrowatch'a (и отнюдь не при десяти участниках), а в 2008 году поднялся на третью ступень пьедестала почёта. Которую покинул только в 2011 году — и лишь для того, чтобы взойти на ступень первую. На которой LM и пребывает по сей день, причём со всё более увеличивающимся отрывом от ближайшего соперника — а им почти бесспорно выступает прародительский Ubuntu.

А вот тот факт, что проект Mint на самом деле развивает две линии, рассматриваемые разработчиками как самостоятельные дистрибутивы,

не столь известен. Всё сказанное о положении в рейтинге относится к собственно Linux Mint (далее просто LM). Который основан на кодовой базе Ubuntu, опирается на его официальные репозитории и полностью совместим с репозиториями неофициальными, так называемыми PPA. И отличается от прародителя, во-первых, официально поддерживаемыми графическими средами, в качестве которых выступают Cinnamon, собственная разработка проекта Mint, и Mate (редакции с KDE и Xfce официального статуса не имеют). А во-вторых, набором специфичных для дистрибутива фирменных утилит, как для командной строки, так и графических, предназначенных для решения очень широкого круга задач — от управления пакетами до блокировки доменов и размещения pastebin-сообщений.

Второй дистрибутив, развиваемый проектом — Linux Mint Debian Edition (далее LMDE) — далеко не столь популярен, как собрат. По его

«Ответившись от Ubuntu, Linux Mint быстро превзошёл своего родителя.»

названию легко догадаться: в нём под пыльным ковром тех же графических сред (Cinnamon и Mate-редакции), расцвеченных фирменными утилитами, лежит не Ubuntu, как в LM, а твердочка-менный Debian.

Первая версия LMDE, вышедшая в сентябре 2010 года, однако, включала среду Gnome 2, точно такую же, что и существовавший тогда LM 9 Isadora. Он получил номер версии 201009 — в отличие от релизов LM, выпуски LMDE именовались по схеме год-месяц. Это было связано с тем, что

изначально они основывались на кодовой базе Debian Testing, и релизы этого дистрибутива носили «скользящий» характер (так называемые rolling release): сами по себе выпуски официальных образов были просто некими вехами в непрерывном развитии системы. И вслед за первым релизом последовали очередные такие вехи — 201012 и 201109; в последней к Gnome-редакции дистрибутива присоединилась сборка с Xfce — вероятно, в связи с тем, что эта среда получила статус одной из официальных для Debian'a.

Однако в том же 2011 году в развитии LMDE, как и собственно LM, произошёл коренной перелом. С одной стороны, Ubuntu перешёл на использование собственной среды Unity, с другой же — вышел Gnome 3, а развитие Gnome 2 прекратилось, как и поддержка этой линии разработчиками.

Обе новые среды оказались неприемлемыми как для собственно LM, так и LMDE: принципы их построения резко отличались от концепции классического рабочего стола. Кроме того, разработчики Gnome 3 пошли по пути существенного ограничения настраиваемости среды. А среда Unity, в которой без сторонних утилит с настраиваемостью дело обстояло тоже не блестяще, кроме того, не поддерживалась никем, кроме собственно Ubuntu. Как, впрочем, по ряду причин (в том числе и юридических, в связи с требованием для сторонних разработчиков отказа от имущественных прав на свой же код) и по сей день никем, кроме прямых клонов Ubuntu, не поддерживается — для LMDE, основанного на Debian, это было ещё более критично, чем для собственно LM.

В результате некоторых коллизий, на которых я не буду сейчас останавливаться, так как они относятся к истории проекта Mint вообще, а не его

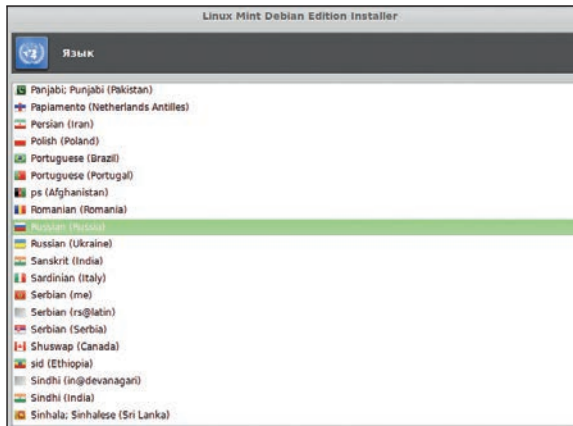


Рис. 1. Языком инсталлятора, предлагаемым по умолчанию, в моём случае оказался русский.

линии LMDE, последняя, начиная с релиза 201205, выходит в двух редакциях — с Cinnamon и Mate в качестве рабочих сред. Поначалу доживала свой век и сборка с Xfce, однако на версии 201205 она и закончила своё существование.

Казалось, что эта судьба уготована и линии LMDE вообще: последний её официальный релиз имел номер 201403, что, как несложно догадаться, соответствовало дате выхода — марту месяцу прошлого года. И затем на протяжении полугода о разработке LMDE не было практически никаких вестей. Пока 28 августа 2014 года на сайте, посвящённом новостям от разработчиков Mint с говорящим названием Segfault (<http://segfault.linuxmint.com/>), не появилось сообщение основателя и бессменного лидера проекта Клемана Лефевра [Clement Lefebvre], известного также как Clem, суть которого сводилась к следующему:

- » все вышедшие до того времени релизы дистрибутива объединяются в версию LMDE 1;
- » версия, находящаяся ныне в разработке, будет именоваться LMDE 2 Betsy;
- » подобно LM, она переходит на фиксированный релиз-цикл и будет основана на кодовой базе следующей версии Debian 8, известной как Jessie, когда та обретёт статус релиза (в тот момент, как и сейчас, она находилась в стадии тестирования);
- » это знаменательное событие произойдёт предположительно в ноябре месяце текущего года.

Последнее обещание выполнено не было. И не из-за нерадивости разработчиков LMDE, а скорее вследствие перфекционизма разработчиков Debian, обычно не выпускающих релизы до полного их «вырелизывания». А сообщения о разработке Betsy опять пропали со страниц Segfault'a почти на полгода — до 6 февраля текущего года, когда Клеман в статье под названием «About Betsy» обрисовал текущее положение дел, а также описал отличия Betsy, с одной стороны, от собственно LM, с другой — от предыдущих выпусков LMDE.

Примерно в это же время на сервере проекта появился, наряду с репозиторием LMDE 1 Debian, и репозиторий LMDE 2 Betsy. А значит — и возможность ознакомиться с будущим релизом до его выпуска, который на сей раз был запланирован на март месяц 2015 года. Как? Расскажу в следующей части этой статьи.

Обретение Betsy. Шаг первый

Никаких директивных указаний о том, как следует использовать репозиторий LMDE 2 Betsy (и даже — следует ли его использовать вообще) на официальном сайте проекта на момент сочинения этих строк ещё не было. Не было и никаких пре-релизных образов будущего дистрибутива. Однако, исходя из общих соображений и солдатской смекалки, последовательность действий представить себе не сложно.



Рис. 2. Горы, равнины и часовой пояс.

Как говорят французы, чтобы сварить суп из курицы, надо как минимум иметь кошку. Поэтому для обретения Betsy необходимо как минимум иметь образ последнего релиза LMDE 1. Каковой, именован **linuxmint-201403-cinnamon-dvd-64bit.iso** и объёмом 1,4 ГБ, можно скачать с любого из зеркал проекта (например, Яндекс). Кроме него, имеется и образ **linuxmint-201403-mate-dvd-64bit.iso**, а также 32-битные сборки обеих редакций; однако меня интересовал именно 64-битный вариант (по причине наличия машины этой архитектуры и 16 ГБ памяти) и с Cinnamon в качестве рабочего окружения (чему причиной — исключительно личные симпатии к этой среде).

Образ диска может быть записан на оптический носитель, на USB-флэшку или SD-карту (одной из специальных графических утилит или прямой командой **dd**). И после загрузки с него мы оказываемся в Live-среде, точно такой же, какая будет после установки, так что сейчас я задерживаться на ней не буду.

Следующий шаг обретения Betsy — установка LMDE 1 как основы дальнейших действий. Описывать детали установки системы, для которой не вышла ещё даже пре-релизная версия, смысла не имеет — в релизе инсталлятор может и измениться. Поэтому остановлюсь только на нескольких моментах, привлекающих внимание.

Аналогично с инсталляторами всех клонов Debian, установка начинается с выбора языка инсталлятора, который в дальнейшем станет языком установленной системы. Как ни странно, в моём случае языком, предлагаемым по умолчанию, оказался русский. Не телепатически ли были определены мои предпочтения?

Очень симпатично выполнена картинка, на фоне которой происходит выбор часового пояса. Поскольку язык инсталлятора был ранее определён как русский, то и часовой пояс соответствует московскому времени.

Критически важный момент установки — выбор раскладки клавиатуры. По умолчанию предлагается русская раскладка в варианте winkeys. Однако можно выбрать и любой другой вариант

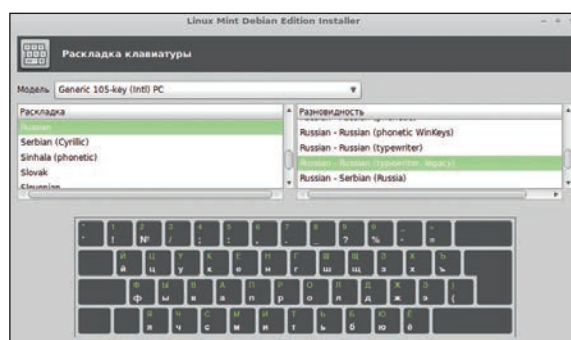


Рис. 3. Выбор вариантов русской раскладки широк, но прибегать к нему не нужно: выбранная раскладка не дополняется к английской, а подменяет её.

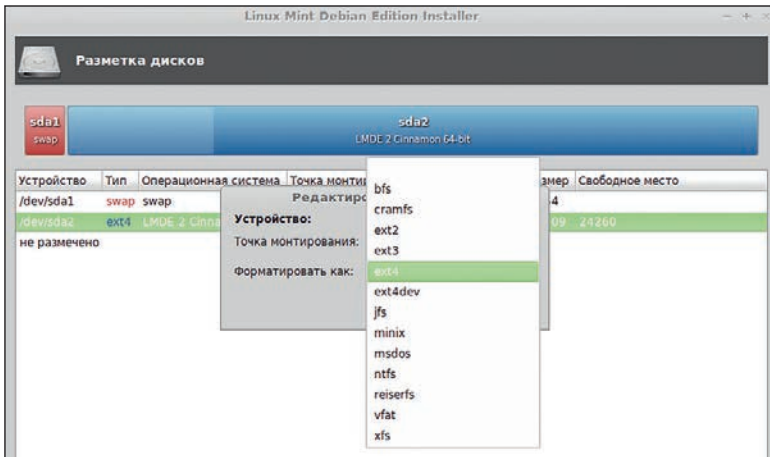


Рис. 4. Список доступных при инсталляции файловых систем не очень обычен.

русской раскладки, причём с возможностью не только проверить её в тестовом поле, но и посмотреть воочию. Правда, делать этого ни в коем случае не нужно — после этого переключиться на английскую раскладку не получится, а без неё невозможно будет заполнить поля учётной записи. Так что лучше сразу остановиться на раскладке English (US).

В ходе подготовки диска к инсталляции, то есть его разметки и форматирования целевого раздела (или разделов), необходимо, как обычно, выбрать для него файловую систему. И в списке предложений, кроме стандартных вариантов (то есть всех файловых систем, нативных для Linux, от btrfs до xfs), привлекают внимание специфическая `cramfs` и постепенно вымирающая `reiserfs`.

Важный момент инсталляции — определение места для загрузчика системы, в качестве которого выступает `Grub 2`. В Ubuntu и всех его производных (в том числе и в LM) для его установки

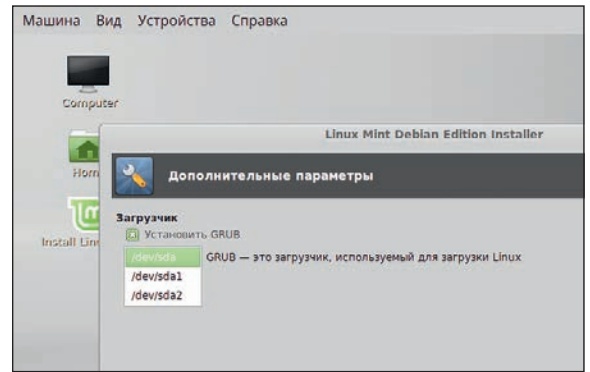


Рис. 5. Инсталлятор LMDE позволяет не только выбрать любое устройство для установки загрузчика, но и отказаться от его установки вообще.

его не туда (например, в MBR первого диска вместо PBR целевого раздела) — это типичная болезнь применителей Ubuntu и его производных, причём не только начинающих (автор этих строк ей тоже подвержен).

Развёртывание системы начинается после нажатия кнопки `Установить` в сводной панели, проходит достаточно быстро и завершается, как всегда, предложением перезагрузить машину, причин не соглашаться с которым я не вижу.

Обретение Betsy. Шаг второй

После перезагрузки мы оказываемся в системе с ядром 3.11, средой Cinnamon версии 2.0 и другими компонентами, актуальными для начала 2014 года. Набор приложений оказывается практически идентичным таковому в LM 16 Petra, одновозрастном нашему LMDE. Поэтому задерживаться на особенностях установленной системы я не буду. Ведь предстоит второй шаг на нашем тернистом пути: волшебное превращение его в LMDE 2 Betsy.

Оно начинается с подключения нужных репозиторий — базового от Debian Jessie и специфического от проекта Mint. Для этого достаточно в каталоге `/etc/apt/sources.list.d/` создать файл `official-package-repositories.list` и внести в него такие строки:

```
deb http://packages.linuxmint.com betsy main upstream import
deb http://extra.linuxmint.com betsy main
deb http://ftp.ru.debian.org/debian/ jessie main contrib non-free
deb http://security.debian.org/ jessie/updates main contrib non-free
deb http://www.deb-multimedia.org jessie main non-free
```

Затем командой

```
$ apt update
```

выполняется обновление локального кэша пакетов, после чего производится тотальное обновление системы:

```
$ apt dist-upgrade
```

В первый раз эта процедура (почти) наверняка завершится сообщением об ошибке при разрешении зависимостей. Ничего страшного: выполняется «коррекция» системы командой

«Выводится панель, наглядно резюмирующая все сделанные при подготовке настройки.»

можно выбрать любое наличное устройство — MBR одного из дисков (если их больше одного) или PBR любого раздела. Нельзя лишь одного — отказаться от установки загрузчика вообще, что иногда бывает весьма желательным. А вот в LMDE это вполне допустимо (как и в его «подковровом» Debian'e).

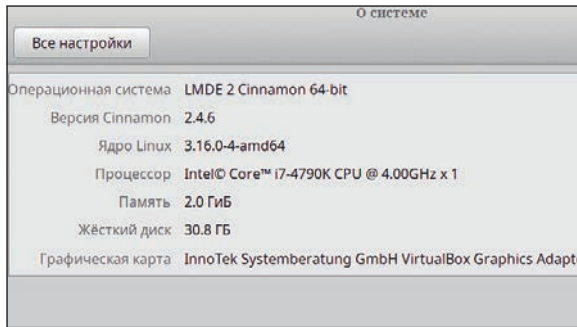
И, наконец, мелкая, но очень существенная деталь: после определения, как надо разметать и форматировать диск и куда надо записывать загрузчик, но до того, как эти действия свершатся в реальности, выводится панель, наглядно резюмирующая все сделанные при подготовке к инсталляции настройки. К ней надо отнестись со всем вниманием, ибо это последний шанс что-то исправить. Например, место размещения загрузчика. Ведь записать

Личная особенность

Читатель, знакомый с системами пакетного менеджмента Debian и его производных, наверняка обратит внимание, что команда `apt` в нашем случае даётся от имени обычного пользователя и не предваряется командой `sudo`. Это не ошибка автора или его лень, а особенность реализации утилиты `apt` в дистрибутивах проекта Mint: она извлекает применителя от необходимости помнить о получении административных прав, так как запрос пароля для авторизации последует за командой `apt` автоматически.



Рис. 6. Сводная информация по подготовке к инсталляции — последний шанс что-то исправить.



► Рис. 7. Атрибуты обновлённой системы.

```
$ apt install -f
```

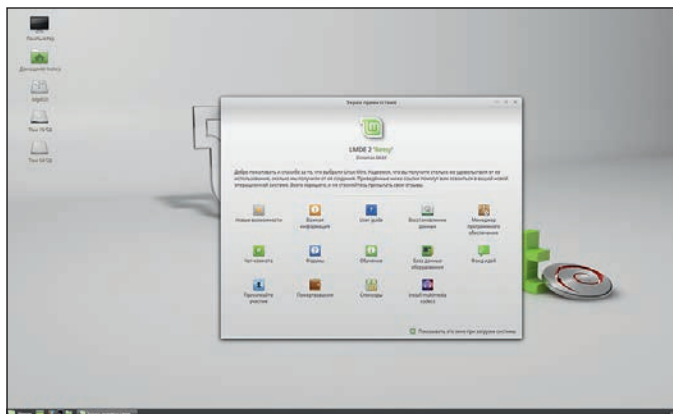
А затем — вторая попытка тотального апгрейда, которая на сей раз наверняка завершится успехом, закрепляемым перезагрузкой системы и лицезрением результата проделанной работы с помощью вызванных из главного меню Cinnamon Системных настроек и их модуля О системе. Который продемонстрирует нам имя обновлённого дистрибутива (LMDE 2), версию его ядра (3.16.0-4-amd64) и среды (2.4.6).

Нельзя сказать, что новообретённая Betsy работает абсолютно гладко. Но ожидать такого от системы, для которой даже пререлиза официально объявлено не было, было бы опрометчиво. И надо заметить, что шероховатости в ней, во-первых, немногочисленны, во-вторых, легко ликвидируются. Но об этом разговор будет позднее. А пока поглядим на дистрибутивоспецифические особенности системы, которые и определяют её лицо. Их три: «ковёр» — рабочая среда, средства её «крепления» — фирменные утилиты, и «подковёрный» Debian.

Среда Cinnamon

Рабочая среда — первое, что видит большинство применителей при запуске системы. Поэтому с неё и начнём знакомство с LMDE 2 Betsy. Правда, во всех дистрибутивах этого семейства среда при первом запуске почти целиком загорожена экраном приветствия проекта Mint, но от него легко избавиться — на время сеанса или навсегда. А потом, при необходимости, запускать, ибо это обычная Mint-утилита.

По отключении экрана приветствия видно, что интерфейс Cinnamon построен в традиционном стиле, включая собственно рабочий стол и управляющую панель. В отличие от Gnome Shell'a или Unity, здесь сразу ясно, что делать дальше. Во-первых, можно щёлкнуть правой кнопкой мыши по рабочему столу, чтобы увидеть его контекстное меню — через него, в частности, можно поменять фоновую картинку или включить слайд-шоу из соответствующего набора.



► Рис. 8. Рабочая среда Cinnamon и экран приветствия проекта Mint.

Привет, мир

Во всех дистрибутивах проекта Mint экран приветствия один и тот же. И кроме формальной информации, он несёт некоторую функциональную нагрузку. Так, он предоставляет простой доступ к Менеджеру программ, базе данных оборудования, чат-комнате (она потребуется для регистрации в качестве члена сообщества, что может быть не лишним — например, для сочинения отзывов о пакетах в Менеджере программ), может обеспечить установку

мультимедиа-кодеков (если опция их установки не была включена при первичной установке системы). Так что, хотя при первом запуске системы экран приветствия, скорее всего, будет отключён, но потребность вызвать его может появиться в дальнейшем. И реализовать эту потребность очень просто: либо через секцию Параметры главного меню Cinnamon, либо прямой командой `mintwelcome` в терминале или мини-терминале.

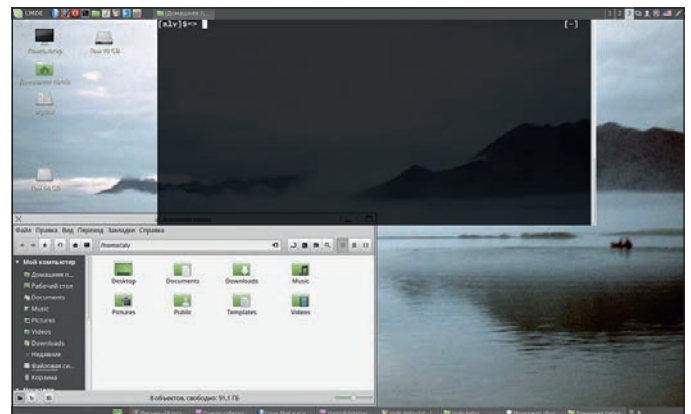
Во-вторых, щёлкнув правой же кнопкой мыши по свободному полю управляющей панели, можно заняться её настройками — например, переместить её на верхний край экрана; или даже установить две панели — верхнюю, с главным меню, пиктограммами запуска приложений, системным лотком и так далее, и нижнюю, на которую удобно поместить апплет со списком всех открытых окон со всех рабочих столов.

Наконец, в-третьих, для доступа ко всем штатным приложениям и утилитам дистрибутива, развёрнутым при первичной установке системы, можно обратиться к главному меню Cinnamon. И это самый главный и самый примечательный элемент интерфейса. Оно вызывается или кнопкой на главной панели, или горячей клавишей (по умолчанию — Left Win, она же Super). В нём можно выделить три элемента — колонку пиктограмм быстрого запуска, поле поиска, поле списка секций меню и поле элементов.

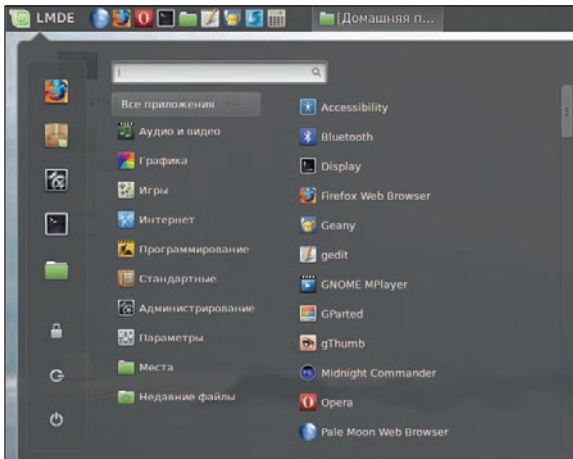
Назначение колонки быстрого запуска понятно: она предназначена для быстрого запуска приложений (смайлики по вкусу), а также блокировки экрана, завершения сеанса или смены пользователя, перезагрузки или выключения машины. Назначение каждой пиктограммы объясняется подсказкой в правом нижнем углу меню.

В поле поиска примечателен инкрементный характер: набор нескольких символов выводит в поле элементов меню только совпадающие значения. Причём совпадения отыскиваются не только в именах программ, но и в их кратких описаниях. Благодаря этому возможен поиск нужной программы и по русским ключевым словам, если её описание имеется в переводе (а для многих приложений дистрибутива это так и есть).

Что же касается полей списков секций и элементов, их основное назначение очевидно — выявление и запуск нужной программы. Хотя первую операцию, на мой взгляд, удобнее делать через поле поиска, потому что русскоязычные названия элементов



► Рис. 9. В среде Cinnamon возможен и двухпанельный интерфейс.

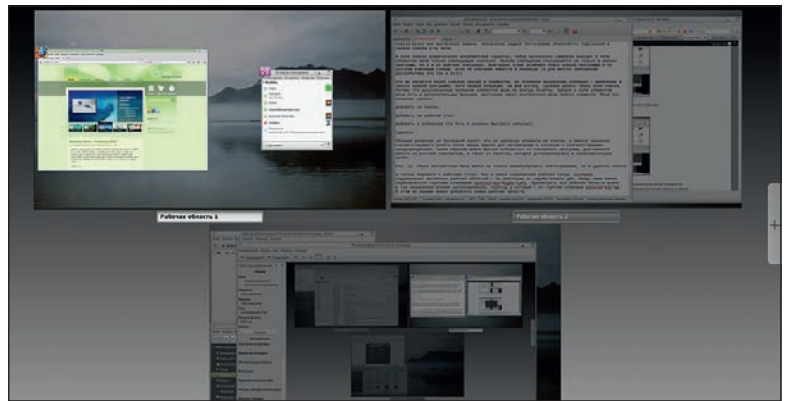
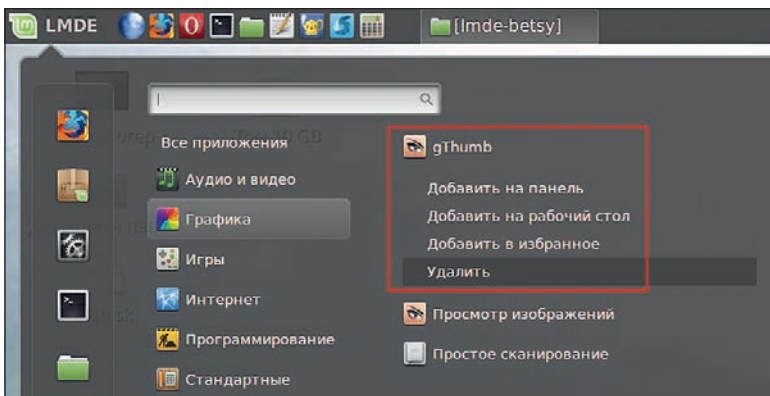


► Рис. 10. Cinnamon: главное меню.

Скорая помощь

Кроме нормального режима, в котором происходит работа приложений, и режима Экспресс, в Cinnamon имеется режим масштабирования, в котором на экран выводятся все открытые окна текущей рабочей области. Переключение в него — горячими клавишами Control+Alt+Up.

► Рис. 11. Через контекстное меню можно не только манипулировать пиктограммами, но и удалять пакеты.



► Рис. 12. Режим Экспресс: с помощью знака «+» на панели справа можно добавлять рабочие области.

меню не всегда понятны. Однако у поля элементов меню есть и дополнительные функции, доступные через контекстное меню любого элемента. Меню это включает пункты:

- » Добавить на панель;
- » Добавить на рабочий стол;
- » Добавить в избранное (то есть в колонку быстрого запуска);
- » Удалить.

Обращаю внимание на последний пункт: это не удаление элемента из списка, а именно удаление соответствующего пакета после ввода пароля для авторизации и согласия с соответствующим предупреждением. Таким образом можно быстро избавиться от «ненужных» программ, доставшихся вместе со штатным комплектом, а также от пакетов, которые устанавливались в ознакомительных целях.

А теперь вернёмся к рабочему столу. Как и любая современная рабочая среда, Cinnamon поддерживает несколько рабочих областей — по умолчанию их задействовано две. Между ними можно переключаться горячими клавишами Control+Alt+Right/Left. Просмотреть все рабочие области можно в так называемом режиме Экспресс, переход в который — по горячим клавишам Control+Alt+Up. В этом же режиме можно добавлять новые рабочие области.

Однако коронка среды Cinnamon — так называемый тайлинг, то есть расщепление экрана на ряд независимых областей, в каждой из которых локализуется окно с запущенным в нём приложением. Это подобно покрытию пола кафелем [tiling], чем и порождена аллюзия. При этом понятие управления окнами как бы лишается смысла — тайлинговые системы управляют не столько окнами, сколько областями экрана, в которых окна открываются. В Cinnamon тайлинг управляется перетаскиванием окон к границам экрана или его углам, а также с клавиатуры — клавишей Super в комбинации со стрелками управления курсором. Поддерживается расщепление экрана пополам по вертикали и горизонтали, а также на «четвертинки».

Отдельно надо сказать о настройках среды Cinnamon. Почти все они выполняются через единый Центр управления (он же — Системные настройки), модули которого охватывают большинство пользовательских потребностей в индивидуализации системы.

Завершая разговор о среде Cinnamon, ещё раз подчеркну: в ней в оптимальных дозах сочетаются традиции, такие как общий интерфейс рабочего стола, и модерн — инкрементный поиск в меню (восходящий к Dash из Unity), режим Экспресс для рабочих областей, тайлинг окон. Что немаловажно, почти все настройки среды можно выполнить её собственными штатными средствами, не прибегая к сторонним твикерам, без которых не обойтись в Unity и GNOME Shell.

Фирменные утилиты

Комплект утилит, специфичных для дистрибутивов проекта Mint — второй фактор, определяющий их своеобразие. Как уже было сказано, утилиты эти покрывают весьма широкий круг задач и потому весьма многочисленны — полный их список включает более 20 исполняемых файлов вида *mint**. Большинство из них имеет графический интерфейс и запускается из секции Администрирование главного меню. Текстовые утилиты, разумеется, требуют запуска из командной строки терминала или консоли.

Важнейшим из фирменных инструментов, безусловно, является собственная реализация утилиты *apt* — именно она обеспечивает удобство и простоту управления пакетами, не превзойдённую во всём семействе Deb-дистрибутивов. Однако ей будет посвящена специальная статья. Здесь же я остановлюсь только на основных графических инструментах. Рассмотрение которых целесообразно начать с Менеджера программ, как наиболее востребованного его компонента.

Менеджер программ *mintinstall* принадлежит к классу самых «высокоуровневых» инструментов для управления пакетами, которые можно назвать интегрированными центрами приложений, поскольку первым их представителем был Центр приложений Ubuntu. В отличие от последнего, *mintinstall* обладает минималистичным дизайном без всяких баннеров и прочих украшательств: только поле поискового запроса, пиктограммы категорий программ и строка состояния текущих действий.

В обращении *mintinstall* столь же прост, как и внешне. А обращение с ним, разумеется, начинается с поиска нужного пакета. Проще всего для этого воспользоваться полем поискового запроса, где вводится имя пакета, его фрагмент или ключевое слово из описания (в последнем случае иногда можно даже по русски). Правда, порядок вывода пакетов — не по соответствию имени, а по количеству отзывов на них.

Если дважды щёлкнуть на строке с именем найденного пакета, появится страница с его описанием, иллюстрирующими

скриншотами, перечнем зависимостей, а также отзывами о пакете — теми самыми, которые определяют приоритет вывода при поиске. Для установки отсутствующего пакета (вместе с его зависимостями) достаточно нажать соответствующую кнопку, для удаления пакета установленного — заменяющую её кнопку Удалить.

А можно также и оставить свой отзыв. Правда, для этого надо предварительно зарегистрироваться в сообществе пользователей (как — расскажу позже).

Определившись, путём чтения описания и, возможно, отзывов (хотя они очень редко несут какую-либо информацию, кроме эмоциональной), с нужностью найденного пакета, его остаётся только установить. Для чего требуется нажать соответствующую экранную кнопку — и процесс начнётся без единого вопроса. Нужно только учитывать, что пакеты, установленные как зависимости удаляемого, удалены не будут, как и в случае с удалением пакетов через главное меню Cinnamon.

С Менеджером программ тесно связана утилита *software-sources*, которая в главном меню именуется Источники приложений. Она предназначена для выбора зеркал основных репозиториях, подключения репозиториях дополнительных, проверки подлинности ключей, исправления проблем в списках пакетов и очистки от следов жизнедеятельности, то есть локальных кэшей.

Сюда же примыкает Менеджер обновлений *mintupdate* — средство для обновления системы. По умолчанию он включается в автозагрузку; пиктограмма его находится постоянно в системном лотке, изменяя свой вид в зависимости от доступности обновлений.

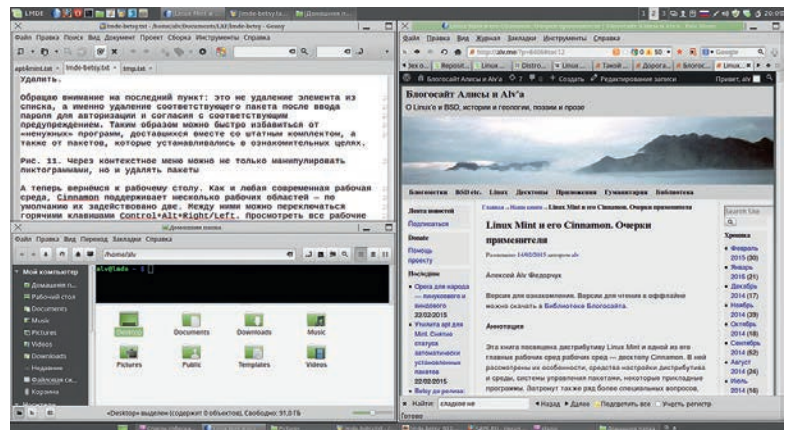
При доступности обновлений получить визуальное представление о них можно, щёлкнув мышью на пиктограмме. После этого будет выведен список пакетов, которые могут быть обновлены в данный момент времени. Строго говоря, вывод не совсем по пакетный: в одной строке списка может быть сгруппировано несколько родственных пакетов, которые друг без друга всё равно не устанавливаются, например, *cinnamon* и *cinnamon-common*. Эту группировку не следует путать ни с зависимостями, ни с метапакетами — она делается исключительно для компактности представления и лёгкости восприятия.

Режима автоматического обновления в Mint не предусмотрено как класса — оно начинается только после нажатия экранной кнопки Установить обновления. Однако некоторые пакеты, критически важные для функционирования системы (ядро, *glibc* и некоторые другие), не обновляются и в таком полуавтоматическом режиме — это требуется задать явным образом, отметив соответствующий бокс.

От средств работы с пакетами плавно переходим к средствам работы с файлами. А тут одно из важнейших дел — резервное копирование. Для чего в составе фирменного инструментария Mint имеется утилита *mintbackup* (в меню — Резервное копирование). Она выполняет две функции: собственно резервное копирование и сохранение списка установленных пакетов.

Для решения первой задачи указываются исходный и целевой каталоги, а также дополнительные параметры — простое копирование; архив *tar*, *tar.bz2* или *tar.gz*; условия перезаписи. Далее можно определить исключения из исходного каталога, не подлежащие архивированию (если, конечно, они нужны). В общем, в этой утилите нет ничего такого, чего нельзя было бы сделать с помощью утилиты *tar* и её опций. Но всё это представлено в наглядной форме, избавляющей от необходимости ломать голову над последними.

Не менее проста и полезна утилита *mintbackup* во второй своей части, сохраняющей список установленных пакетов. Здесь всего-то и требуется, что указать целевой каталог, при желании — просмотреть список пакетов, установленных в системе,



► Рис. 13. Тайлинг окон в среде Cinnamon: расщепление экрана на «половинки» и «четвертинки».

нажать кнопку Применить и дождаться появления сообщения об успешном завершении процесса. После этого в целевом каталоге обнаруживается обычный текстовый файл вида `software_selection_[имя хоста]@[data]-package.list` со списком пакетов.

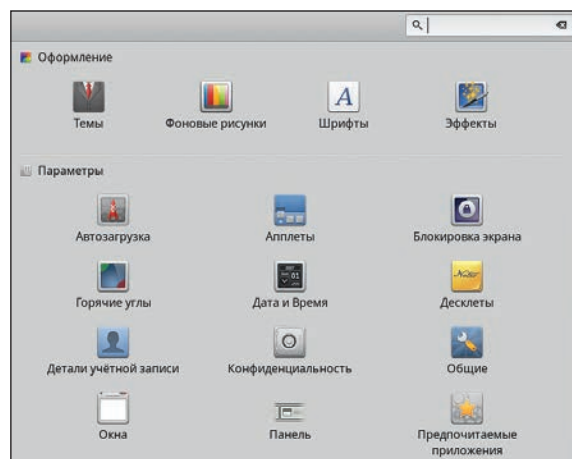
Как известно, на смену оптическим приводам постепенно приходят USB flash и SD-карты. Единственная сфера, где до некоторого времени оптические накопители были не всегда заменимы — это установка системы на чистую машину. Однако ныне все современные дистрибутивы Linux'a или BSD-системы распространяются в виде так называемых гибридных образов, допускающих их запись на твердотельные носители. Что повлекло за собой появление большого числа программ, призванных выполнить эту процедуру.

Имеется такая утилита и в составе фирменного инвентаря Mint'a. Это *mintstick*, которая в главном меню находится в разделе Стандартные под именем Создание загрузочного USB-носителя. Действия в ней очевидны: надо выбрать записываемый образ и указать, куда он должен быть записан (подключённая флэшка или SD-карта предлагается по умолчанию). После этого остаётся ввести пароль и подождать завершения процесса.

В Системных настройках Cinnamon имеется модуль Языки. Однако он не является частью этой среды, а принадлежит к семейству фирменных утилит Mint (под именем *mintlocale*). Он выполняет двойную функцию. Во-первых, через него можно изменить собственно язык интерфейса и прочие параметры, объединяемые понятием locale (формат даты, представление чисел, единицы измерения etc.). Главная его особенность — разделение групп параметров Language и Region (в русском переводе — Язык и Область/Край, соответственно). Первая определяет переменные LANG и LC_TIME, то есть собственно язык интерфейса

Скорая помощь

К низкоуровневому конфигурированию с помощью редактора *Dconf* в среде Cinnamon приходится прибегать буквально в единичных случаях — настолько редких, что соответствующий инструмент даже не включён в штатную поставку дистрибутива, а должен устанавливаться дополнительно. Пожалуй, единственный пример того, когда без него не обойтись — включение режима сохранения сеанса при завершении работы (см. рис. 15).



► Рис. 14. Среда Cinnamon: системные настройки включают модули для конфигурирования почти всего, чего можно.

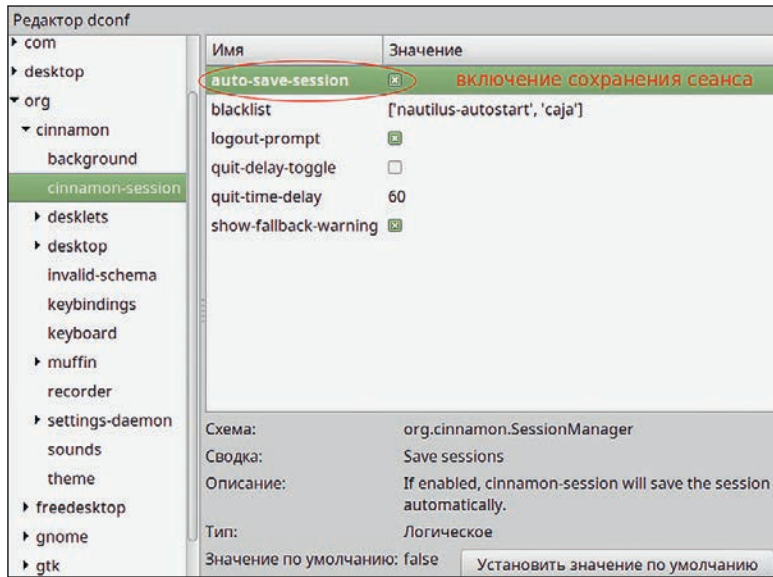


Рис. 15. Сохранение сеанса — один из очень редких случаев, когда нужен редактор *Dconf*.

и местное время. Группа же параметров Region задаёт значения всех остальных локально-зависимых переменных — представленные чисел и денежных единиц, формат бумаги и так далее.

Разнесение переменных по группам может показаться спорным, как и его аргументация Клеманом («специально для отъезжающих за границу»). Однако сама по себе идея отделения языка интерфейса от прочего локально-зависимого хозяйства весьма здрава, особенно если язык этот — смесь нижегородского с оксфордским, как часто можно видеть в дистрибутивах, адаптированных для наших палестин.

Вторая функция *mintlocale* — определение так называемого метода ввода [Input Method], то есть системы обеспечения ввода символов, отсутствующих на клавиатуре от слова «вообще». На-

«Идея отделения языка интерфейса от прочего локально-зависимого хозяйства здрава.»

пример, китайских иероглифов и символов всех генетически связанных с ними систем письма. Конечно, жизненно необходимых жителям соответствующих стран, но у нас актуальных только для применителей со специальными интересами.

Говорить о Mint-утилитах можно было бы ещё долго, потому как я не описал и половины из них. Однако для плавного перехода к следующей части закончу эту главу о том, какой утилиты в LMDE, в отличие от собственно LM, нет: Менеджера драйверов, предназначенного для управления проприетарными программами

поддержки оборудования. Что, однако, не единственное различие между этими дистрибутивами, как мы увидим в следующем разделе.

LM и LMDE: Сравнение

Всё сказанное о среде Cinnamon и фирменных утилитах имеет силу как для LM, так и для LMDE (с единственной оговоркой относительно Менеджера драйверов). В чём же разница между этими дистрибутивами? Рассмотрим этот вопрос с позиций применителя, не горящего желанием залезать «под ковёр» (а тем более «под капот») без большой необходимости.

Так вот, под данным углом главное различие между LM и LMDE — отсутствие во втором доступа к PPA-репозиториям Ubuntu, который в первом возможен без проблем. Попытка в LMDE подключить через *add-apt-repository* какой-нибудь из репозитория с Launchpad'a вызывает сообщение об ошибке:

Добавление PPA не поддерживается

С другой стороны, для применителей LMDE доступны как официальный репозиторий Debian, так и неофициальные, которых тоже немало количество. Конечно, той централизации, что реализована в дистрибутивах семейства Ubuntu через Launchpad, нет. Но и оснований жаловаться на бедность (приложениями) нету тоже.

На практике с проблемами при установке пакетов приходится сталкиваться только в редких и достаточно специальных случаях. Причём часть этих проблем связана с тем, что LMDE — не совсем LM, но и не совсем Debian Jessie, например, при установке модулей поддержки системы размещения данных ZFS. Как уже было сказано, подключить PPA-репозиторий, из которого эти модули устанавливаются в LM, здесь не удастся. А попытка «в лоб» воспользоваться рецептом, приводимым на сайте проекта ZFS on Linux для «чистого» Debian приводит к ошибке при создании файла описания репозитория, ибо он явным образом предназначен для Jessie. Что, впрочем, обходится тем, что соответствующий файл */etc/apt/sources.list.d/zfsonlinux.list* создаётся вручную, и в него вносятся такие строки:

```
deb [arch=amd64] http://archive.zfsonlinux.org/debian jessie main
deb-src http://archive.zfsonlinux.org/debian jessie main
```

В итоге после выполнения команды *apt update* все пакеты поддержки ZFS благополучно собираются и устанавливаются.

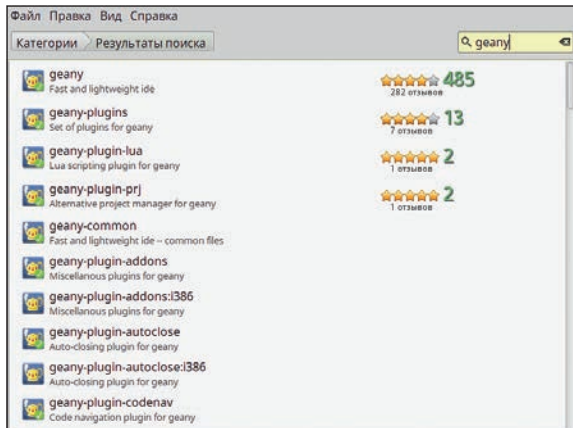
В других случаях проблемы возникают вследствие, видимо, до-релизного статуса Betsy. В частности, у меня они возникли при установке систем кастомизации образов дистрибутивов. К этому занятию я пристрастился за время применения LM, где применял в этих целях пакет *UCK* (то есть Ubuntu Customization Kit, работа которого основана на добавлении и удалении пакетов из имеющегося образа), входящий в официальный репозиторий одноимённого дистрибутива. Как и следовало ожидать, в LMDE этот пакет установить было невозможно. Однако, и тоже ожидаемо, *UCK* отказался работать с его образом, будучи запущенным из LM. Мотивируя тем, что ни малейших следов Ubuntu он в этом образе не обнаружил.

Для сборок кастомизированных образов имеются и другие средства; самое из них известное — пакет *remastersys*. Он основан на совсем другом подходе, нежели *UCK* — на создании снапшота установленной, настроенной и «отрихованной» по набору пакетов системы; но позволяет получить сходный с ним результат.

В LM пакет *remastersys* благополучно устанавливается из PPA-репозитория и работает безукоризненно. В LMDE же дело упирается в то, что разработчик пакета пару лет прекратил свою деятельность, остановившись в своих сборках на версиях для Ubuntu и Debian того времени (а буквально на днях и сайт его прекратил работать в связи с истечением срока делегирования домена). Сборки для Ubuntu с тех пор поддерживаются в актуальном состоянии майнтейнером соответствующего PPA-репозитория.

Рис. 16. Менеджер программ — простой способ установки и удаления пакетов.





► Рис. 17. Поиск пакета *Geany* и его плагинов в Менеджере программ.

Авторская же сборка для Debian (то есть для Wheezy) в Betsy, си-речь в Jessie, устанавливаться отказывается наотрез: с тех пор изменились не только версии пакетов-зависимостей, но и их набор.

Впрочем, выход нашёлся и здесь — в виде форка *remastersys*, пакета *refractasnapshot*. Хотя и его установка не обходится без проблем, связанных с разрешением зависимостей, которое в конечном счёте приходится выполнять вручную. Однако завершается это дело благополучно.

Существует мнение, что в репозиториях Debian, даже ветки Testing, версии пакетов более старые, нежели в официальных репозиториях Ubuntu и, тем более, в PPA-репозиториях. И нередко это действительно так. Но, во-первых, возникает вопрос: а всегда ли действительно так нужны новейшие версии пакетов? Ведь новая функциональность в них нынче появляется достаточно редко, а обновления для устранения критических уязвимостей во всех дистрибутивах выпускаются всеми майнтейнерами, кроме самых нерадивых.

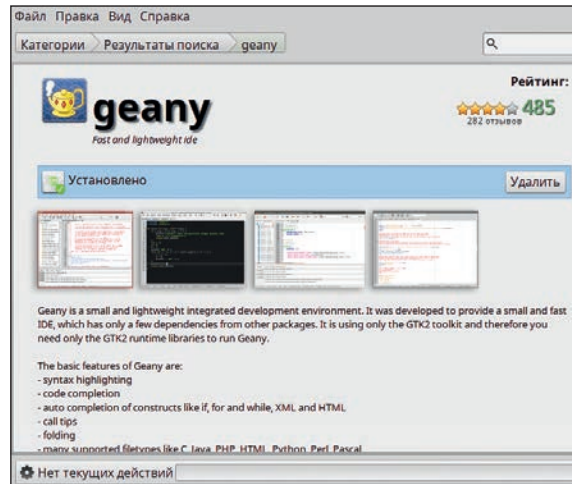
Кроме того, высказанное выше расхожее мнение имеет множество исключений. Например, в момент сочинения этих строк в официальном репозитории Debian Jessie (доступном, разумеется, и в Betsy) пакет *calibre* представлен версией 2.5, тогда как в Ubuntu — только 1.48. А ведь это один из тех нечастых случаев, когда различия старой и новой версии — не в косметике, а в функциональности.

И к тому же пакеты, принципиально важные именно для дистрибутива LMDE, как правило, поддерживаются для неё в официальном репозитории Mint — и вот там-то их версии уже точно самые последние.

Следующее различие между LM и LMDE касается аппаратных средств. Как известно, производители оборудования не балуют Linux своим вниманием. Однако если уж хоть как-то реализуют поддержку своей продукции для этой ОС, то обычно для любого её дистрибутива — при условии, сказал бы старик Генри Форд, что это будет Ubuntu. И потому всякого рода фирменные драйвера могут иметься для него (и, соответственно, быть доступными для его производителей, включая LM), но отсутствовать для Debian и LMDE. Не случайно ведь, как было отмечено в конце предыдущего раздела, в LMDE из всего набора Mint-утилит нет только Менеджера драйверов.

Однако отсутствием Менеджера драйверов дело не ограничивается. В начале статьи я упоминал о некоторых шероховатостях в работе «рукотворной» Betsy. Одна из них — отвратительное качество звука сразу после обновления исходной системы. В частности, при использовании HD Audio от Intel его можно определить как нечто среднее между стоном и хрюканьем.

Это решилось совсем просто — созданием файла `/etc/modprobe.d/alsa-base.conf` и внесением в него такой строки:



► Рис. 18. Менеджер программ: описание найденного пакета.

```
options snd-hda-intel vid=8086 pid=8ca0 snooper=0
```

После чего звучание становится пригодным для человеческого восприятия.

А теперь о хорошем. В числе различий между LM и LMDE основной разработчик обоих дистрибутивов, Клеман Лефевр, указывает, что последний дистрибутив проще с точки зрения архитектуры и, как следствие, обычно несколько стройней, быстрее и отзывчивей. Мои наблюдения с этим согласуются: при запуске с традиционного винчестера LMDE показывает большее субъективное быстродействие (которое и можно квалифицировать как отзывчивость), нежели LM: оно визуально сопоставимо скорее с быстродействием последнего при его установке на SSD.

И последнее различие, которое можно констатировать в данный исторический момент: LM, основанный на Ubuntu 14.04 LTS, сохранил верность собственной системе инициализации, *upstart*, тогда как Betsy, вслед за своей «подковёрной» Jessie, перешёл на *systemd*.

Вместо заключения

Настоящее заключение для LMDE Betsy давать ещё рано — время его наступит после выхода релиза и практического применения оного. Однако предварительные итоги можно подвести. Главный из них в том, что LMDE 1, последний раз обновлявшийся около года назад, без всякого труда превращается в пре-релизный Betsy, который, во-первых, вполне пригоден к выполнению любых более или менее стандартных практических задач, во-вторых, отличается завидной устойчивостью, и в-третьих, без проблем обновляется штатными средствами, в частности, через *mintupdate*. А с помощью систем пакетного менеджмента в нём легко устанавливаются любые пакеты из официальных репозиториях как LM, так и Debian'a. Что вселяет надежду на то, что в грядущем релизе всё будет ещё более хорошо.

Предвкушение релиза Betsy омрачается для любителя линии проекта Mint и среды Cinnamon тем, что этот любитель окажется в ситуации Буриданова осла: он будет вынужден выбирать между двумя одинаково вкусными и равно доступными охапками сена.

И уж совсем в заключение я хотел бы выразить признательность Станиславу Шрамко (он же — stanis) за плодотворное обсуждение всех затронутых в этой статье вопросов. **15.03**

Post Scriptum

Работа над статьёй была практически закончена, когда в блоге разработчиков проекта Mint появилось сообщение Клемана Лефевра о начале тестирования LMDE 2 Betsy. И мы успели приложить к LXF DVD этого номера 64-битный LMDE 2 RC Cinnamon! Но интересно будет проделать упражнения, описанные в статье, и сравнить получившийся дистрибутив с официальным.

Скорая помощь

Следует помнить, что поиск в Менеджере программ — регистро-зависимый. Это можно продемонстрировать на примере поиска выпадающих терминалов. Если в поле поиска ввести слово «Выпадающий», мы увидим пакет выпадающего терминала *Guake*. А по ключевому слову «выпадающий» обнаружится совсем другой выпадающий терминал, *Tilda*.

Matuntu: Mate по-русски

Татьяна Иванова представляет дистрибутив на базе Ubuntu, разработанный специально для русскоговорящих пользователей.

Дистрибутив Ubuntu оказался самым продуктивным генератором производных систем — на данный момент на Distrowatch зарегистрировано более 270 дистрибутивов, основанных на кодовой базе Ubuntu (или на Ubuntu и Debian вместе). В причины этого явления вдаваться здесь неуместно, хотя об одной из них будет сказано чуть позже. Цели при создании таких дериватов преследовались самые различные — от реализации личных амбиций в виде «нескучных обоев» до сборки узкоспециализированных систем производственного назначения.

Отдельной группой среди дериватов Ubuntu являются локально-адаптированные системы, то есть собранные с учётом реалий конкретных стран и регионов. Тут с починю и успехом следует поздравить испанцев — чуть ли не в каждой провинции этой страны существует клон или дериват Ubuntu, ориентированный на применение в местных органах управления, разработка которого подчас финансируется из муниципального бюджета.

Однако на Руси, как известно, народ тоже не валенком щи хлебает, так что и у нас появились системы, основанные на Ubuntu и приспособленные к условиям самой большой страны нашей планеты. Правда, в отличие от Испании, ни федеральные, ни региональные органы не проявили пока пламенного желания материально поддержать эти разработки, и они ведутся силами волонтеров-энтузиастов.

Здесь на первое место надо поставить дистрибутив Runtu, созданный в далёком 2007 году и основанный на идее синтеза простоты и доступности Ubuntu с полноценной русификацией, русской локализацией, набором необходимого программного обеспечения и готовностью к применению сразу после установки. Дистрибутив этот переживал и хорошие времена, и периоды затухания разработки, но ныне, по итогам 2014 года, может считаться активно развивающимся. Кроме того, он послужил поводом для создания другого дистрибутива, который и будет объектом дальнейшего рассмотрения.

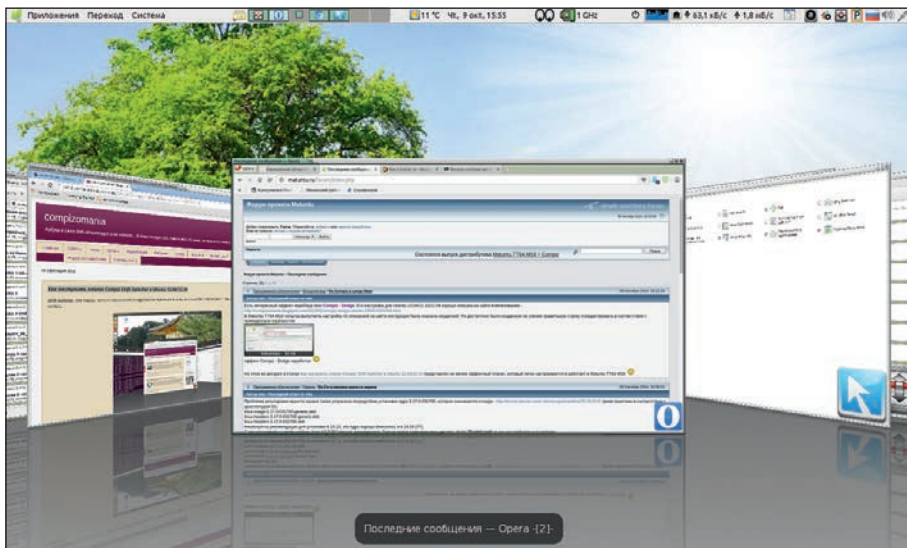
Летом 2012 года в недрах сообщества Runtu началось обсуждение возможности создания дистрибутива на пакетной базе Ubuntu с графическим окружением Mate (форк Gnome2), поскольку Mate в тот период времени в качестве рабочего стола официально не был признан фирмой Canonical. Впрочем, проект Ubuntu Mate и по сей день не имеет официального статуса.

Было проведено много экспериментов с использованием Mate в качестве второго окружения. Но желание, чтобы дистрибутив был только с Mate, способствовало тому, что в результате ровно через год, в июне 2013 года, идея была реализована в виде образа операционной системы с русской локализацией по умолчанию под именем Matuntu. Почему такое имя? Во-первых, Ubuntu-based, во-вторых, графическая оболочка Mate, в-третьих, такое сочетание весьма благозвучно и созвучно собратьям из семейства убунтовых.

Выше было обещано указать об одной из причин активного создания дериватов Ubuntu. Причина эта — наличие нескольких простых в использовании и эффективных по результатам программ для сборки на её основе собственных, индивидуализированных систем. Одна из таких программ, *Remastersys*, и была использована для создания Matuntu.

Remastersys — специальный инструмент (утилита) для создания резервной копии (бэкапа) установленной ОС, опционально — вместе со всеми данными. Созданная резервная копия может быть установлена/перенесена на любой ПК, либо использована в качестве LiveCD. Отличное руководство по применению этой утилиты для создания LiveCD сотворил активный член сообщества Алексей Бухалов, он же ВааТЛТ, предоставивший его во всеобщее пользование.

Кроме русской локализации, ключевым моментом при создании Matuntu была адаптация системы к предпочтениям отечественной части сообщества: например, в сборку включены утилиты *gCDEmu* для монтирования образов, в т.ч. BlueRay, *inxi* — удобная консольная утилита для вывода информации о железе и системе,



Удобный и привлекательный эффект Shift Switcher позволяет переключаться между открытыми приложениями с помощью клавиатурных комбинаций.

Мастер-ремастер

Программа *Remastersys* до некоторого времени присутствовала в сборках для Ubuntu и Debian. Однако три года назад автор прекратил её развитие, и последний поддерживаемый релиз Debian — Wheezy.

А недавно и сайт автора прекратил существование в связи с истечением срока делегирования домена. Так что дальнейшая судьба её не ясна. Сборка же для Ubuntu развивается майнтайнером

соответствующего PPA-репозитория, где поддерживается в актуальном состоянии: пакеты её предназначены для всех текущих релизов Ubuntu и для готовящегося релиза 15.04.

Net Activity Viewer — удобное графическое приложение на *GTK2* для просмотра сетевых подключений, *Boot-Repair* — утилита для восстановления загрузки, *USB Creator* — для создания загрузочной флешки, и много другого полезного программного обеспечения.

В создании Matuntu, кроме автора этих строк (известного также как vita), принимали активное участие члены сообщества Runtu — Игорь Мартынов, он же ivm, Алексей Бухалов, он же ВааТЛТ, и Юрий Квасов, он же salambus, ушедший от нас в июле 2013 года. Первые сборки базировались на основе Ubuntu 12.04 для 32-разрядной архитектуры — Matuntu_A1, Декабрьская Matuntu и Matuntu-Best с Mate 1.6. Позднее, с выходом Ubuntu 14.04 и выпуском Mate версии 1.8.0, были созданы новые сборки Matuntu — Matuntu_Trusty с Mate 1.8 (x32) и Matuntu-TT64 с Mate 1.6, с возможностью дальнейшего обновления Mate (x32_x64).

Matuntu привлекла к себе большое внимание пользователей, и в связи с положительными откликами продолжает совершенствоваться и развиваться. Главным элементом оформления Matuntu стали листики Mate — фавикон сайта, в меню Live-DVD-образа, в меню *Grub*, в теме Plymouth и значке меню на панели.

В настоящее время актуальными являются следующие сборки дистрибутива:

» **Matuntu-PP32-nonpae** — для новых инсталляций на оборудовании, поддерживающем только 32-битную архитектуру, с ядром без pae, т.е. для компьютеров с невысокой производительностью, без установки офиса и *GIMP* и с минимумом установленных интернет-приложений.

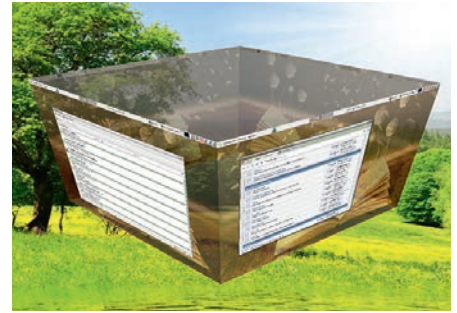
» **Matuntu-PP32-M16+Compiz** — для новых инсталляций на оборудовании, поддерживающем 32-битную архитектуру, с ядром 3.13.0-43-generic, предустановленными композитным менеджером *Compiz* и его эффектами, а также большим набором ПО, в т.ч. уже прекратившим поддержку, например, *gCue2tracks* — утилита для разрезания

образов аудиодисков (копий “lossless”) на отдельные файлы/треки по индексному CUE-файлу.

» **Обновлённая Matuntu-TT64-M-18+Compiz** — для новых инсталляций на оборудовании, поддерживающем 64-битную архитектуру с ядром 3.16.0-28-generic, на пакетной базе Ubuntu 14.04.1 LTS с поддержкой до апреля 2019 года. Автозапускаемый *Compiz* с настроенными эффектами и большой набор ПО. Реализована работа с беспроводными стерео-аудиоустройствами в режиме A2DP.

» **Matuntu-TT32-M18** — для новых инсталляций на оборудовании 32-битной архитектуры, ядро 3.16.0-28-generic. Сборка создана 20 декабря 2014 года на пакетной базе Ubuntu 14.04.1 LTS с поддержкой до апреля 2019 года. В качестве оконного менеджера используется *Marco* с двумя панелями: верхней с Главным меню и нижней, автоскрываемой, с областью уведомлений и апплетом подключаемых носителей.

» **Matuntu-TT64-M16** — для новых инсталляций на оборудовании, поддерживающем 64-битную архитектуру, с ядром 3.16.0-30-generic. Сборка создана 25 января 2015 года на пакетной базе Ubuntu 14.04.1 LTS со стабильной версией графического окружения Mate 1.6.2. В качестве окон-



» **Compiz** запускается по умолчанию и обеспечивает любимые пользователями эффекты.

» **PP** — Precise Pangolin 12.04 (Педантичный Панголин);

» **TT** — Trusty Tahr 14.04 (Надежный Тар).

На мой взгляд, такой подход в наименовании упрощает представление об основных компонентах операционной системы, из которых собираются разные варианты дистрибутива.

Matuntu максимально русифицирована, что привлекательно для российских пользователей, не всегда владеющих иностранными языками. Процесс инсталляции достаточно быстрый. Установка в принципе мало отличается от стандарт-

ной для Ubuntu, но без включения слайд-шоу, выбирается русский язык в начале инсталляции стрелкой вверх в Matuntu-TT, стрелкой вниз в Matuntu-PP, далее выполняются шаги по умолчанию до момента выбора способа установки

(рекомендуется использовать Другой способ, чтобы выбрать заранее подготовленный раздел для инсталлируемой системы), остальное выполняется по интуитивно понятным подсказкам.

Для любителей эффектов композитного оконного менеджера созданы сборки с предустановленным и автоматически запускаемым *Compiz*.

Во всех версиях Matuntu имеется возможность работы с беспроводными аудиосистемами в режиме высококачественного звука A2DP, что весьма актуально для использования современной акустики.

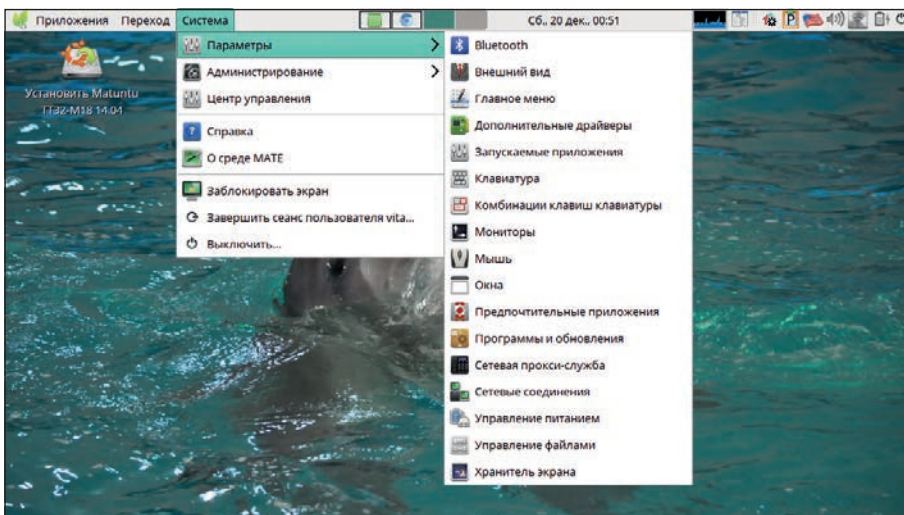
10 июля 2014 года на сайте <http://matuntu.ru/> был создан форум, на котором обеспечивается поддержка пользователей Matuntu, учитываются их пожелания при создании новых сборок.

Таким образом, данный дистрибутив подойдёт для русскоговорящих пользователей, серьёзно рассматривающих возможность перейти с Windows на доброжелательную альтернативную ОС. Потому что Matuntu — это система, в которой всё настроено и работает практически «из коробки». А дружное сообщество Matuntu, сплотившееся на форуме, способно обеспечить действенную помощь в домашнем применении системы. Кроме того, пользователи Matuntu всегда могут получить поддержку сообщества всего семейства дистрибутивов, базирующихся на пакетной основе Ubuntu, а также обратиться к обширной документации при возникновении трудностей или поделиться приобретённым опытом. И это повышает интерес к нашему дистрибутиву. **LXF**

«Возможность перейти с Windows на доброжелательную альтернативу.»

ного менеджера используется *Marco* с двумя традиционно оформленными панелями. Предусмотрен обширный набор приложений различного направления.

Для более чёткого представления различий между сборками Matuntu в последних их выпусках стали применяться условные обозначения, включающие в себя аббревиатуру кодового имени Ubuntu, цифры 32 или 64, указывающие на архитектуру системы, и M — рабочий стол Mate (цифрами указывается версия Mate):



» Matuntu максимально русифицирована, и по умолчанию образ системы в Live-режиме загружается с русской локализацией.



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.

Жалобы луддита

Кое-что для тех, кому, как и мне, кажется, что прогресс длится уже слишком долго...

Жалобы луддита

Пергамент с пером гусиным,
Чернила и нож перочинный —
Не дают они быстро писать,
Но время дают размышлять.

Авторучки мне не симпатичны,
Им нету дороги в мой дом.
Гораздо руке привычней
Пергамент с гусиным пером.

Печатных машинок услуга —
лишь прихоть ленивых людей.
Скучаю по старому другу,
Авторучке верной моей.

Оболочка с командной строкою
неудобны со всех сторон.
Печатать лучше умеет
Реликтовый мой «Ремингтон».

Зачем нам дурацкие мыши?
И Gnome — не для меня.
Вернуться бы в *Bash* старый добрый,
к девяносто третьего дням.

Я Unity не уважаю.
И не люблю этот *Chrome*.
Глядя на них, понимаю,
Насколько хорош был Gnome.

Тачпады ещё появились...
Я их не приемлю, шалишь!
По мне, гораздо удобней
Мой верный курсор и мышь.

И *systemd* всякие эти,
И *Wayland* не приму нипочём.
Пойду и куплю в Интернете
Пергамент с гусиным пером.

chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



Групповуха с фото

Обрабатывайте сотни изображений разом с помощью *phatch*, утилиты групповой обработки фотографий.

Пусть вам надо занести картинку в галерею на сайте, но она не того размера. Не проблема: запускаем *GIMP*, открываем картинку, выбираем Image > Scale Image [Изображение > Масштабировать], вводим новый размер и сохраняем. Ура! Ну, а если нужно сделать то же самое с остальными 274 картинками в папке? Тут не обойтись без групповой обработки.

Еще в **LXF169** я писал об утилите командной строки *ImageMagick*, показав, как, обернув небольшим скриптом, применять ее для данной цели. Еще удобнее для этого утилита *phatch* (сокращение от 'Photo Batch' [групповая обработка фото]). Она обязана быть в репозиториях вашего дистрибутива, и установка будет простой. Так, команда

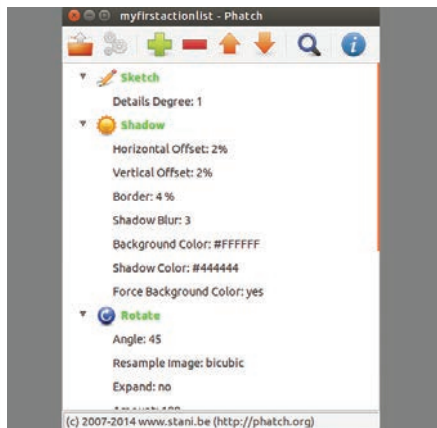
```
$ sudo apt-get install phatch
```

в Ubuntu установит графический интерфейс *phatch* и утилиту командной строки, на которой он работает. Это утилита на Python, и если у вас нет интерпретатора и библиотек Python (что маловероятно), те тоже будут загружены и установлены.

Phatch не предназначен для самой обработки фотографий: с ним, например, не убрать бородавки с портрета вашей бабушки. Он может только составить список действий — очередность операций обработки. Операций много: регулировка контраста и яркости, размывание, определение краев, изменение масштаба, обрезка, поворот, сдвиг перспективы, уменьшение теней, прорисовка контуров и всякие художественные эффекты — карандаш, уголь, размывание... Ключевой финальный этап — сохранение готового изображения, и здесь можно выбрать из большого набора форматов файлов. Создав список действий, можно сохранить его в файл (или снова загрузить список) и выполнить эти действия над папкой с изображениями. Также можно отработать действия по списку, запустив *phatch* из командной строки.

Phatch включает просмотрщик изображений; перетащите изображение в окно, и увидите все метаданные EXIF, которые камера записала в изображение: производителя и модель камеры, разрешение, время и дату, выдержку, диафрагму, ориентацию изображения, баланс белого и т.д. Часть этих параметров даже редактируется.

На подробный рассказ о *Phatch* у меня здесь нет места; установите ее, порезвитесь с ней и воспользуйтесь обучающими руководствами с превосходного сайта <http://photobatch.wikidot.com>.



➤ Создавайте пошаговые списки действий и применяйте их к папкам с изображениями с помощью графической утилиты *Phatch*.

Интерактивные скрипты с Whiptail

Приукрасьте интерактивные скрипты! Создайте текстовый мастер с помощью маленькой умной *Whiptail* и небольшого скрипта.

В мире командной строки программы обычно неинтерактивны: вы указываете всякие параметры в командной строке, нажимаете Enter, а дальше программа работает сама. Но некоторые люди предпочитают, чтобы нужная информация запрашивалась в нужное время — так и появились мастера. Для пошагового сбора информации в скрипте традиционный подход «запросить данные — прочитать данные» может выглядеть так:

```
echo -n "Enter file name to process: "
read FILENAME
```

...и т. д. Но этот подход не лучший с визуальной точки зрения, к тому же в нём непросто реализовать такие вещи, как выбор пунктов меню или установка галочек. Альтернатива — утилита под названием *Whiptail*, которая позволит показывать различные окна для сбора информации из скрипта.

В *Whiptail* несколько типов диалоговых окон (см. таблицу). В большинстве из них есть средства ввода данных — от простого Да/Нет до текста в свободном формате. Для демонстрации некоторых из них я написал простой скрипт; он проводит пользователя через процесс изменения прав доступа на файл или папку и завершается формированием требуемой команды *chmod*. Скрипт не очень практичен и служит лишь для иллюстрации. Вот он. (Номера строк даны для ссылок и не являются частью файла.)

```
1 #!/bin/bash
2
3 FILENAME=$(whiptail --inputbox \
4 "Введите имя файла или папки:" \
5 8 78 --title "Выбор файла" 3>&1 1>&2 2>&3)
6
7 exitstatus=$?
8 if [ $exitstatus != 0 ]
9 then
10  exit 1 # Пользователь вышел
11 fi
12
13 RECURSIVE=»»
14 if [ -d $FILENAME ] # Пользователь выбрал папку
15 then
16  if (whiptail --title "Apply to folder?" --yesno \
17  " Вы ввели имя папки. Хотите применить
18  изменения ко всем файлам в папке?" \
19  8 78)
20  then
21  echo recursive option selected
22  RECURSIVE="-R"
23  fi
24 fi
25
26 MODES=$(whiptail --title "Выбор разрешений" \
27 --checkboxlist \
28 "Выберите разрешение" 10 78 4 \
29 r Read ON \
30 w Write OFF \
31 x Execute OFF 3>&1 1>&2 2>&3 | tr -d -c rwx )
32 exitstatus=$?
33 if [ $exitstatus != 0 ]
34 then
35  exit 1 # Пользователь вышел
36 fi
```

37

38 whiptail --title "Подтверждение" --msgbox \

39 "chmod \$RECURSIVE u=\$MODES \$FILENAME" 8 78

Начну с объяснения, как *Whiptail* взаимодействует со внешним миром. Сперва текстовые данные, введенные пользователем, скажем, в поле ввода, записываются в стандартный поток ошибок. Это немного усложняет жизнь, потому что легко записать эти данные в переменную (используя подстановку команд) или перенаправить их через канал не получится. Через минуту я покажу, как обращаться с этими данными. Затем принятые пользователем решения «да/нет» и «ОК/отмена» передаются обратно через статус завершения программы: при выборе «да» или «ОК» возвращается ноль, а при выборе «нет» или «отмена» — ненулевое значение.

Разберем скрипт. Строки 3–5 описывают поле ввода для получения имени файла от пользователя. Над полем отображается текст 'Выбор файла [File Selection]', а внутри поля — текст-подсказка 'Введите имя файла или папки [Enter file or folder name]'. Числа (8 и 78) задают ширину и длину поля. Всё, что ввел пользователь, заносится в стандартный поток ошибок. Однако причудливый трехшаговый танец 3>&1 1>&2 2>&3 в конце команды рокирует этот поток с потоком вывода, чтобы мы могли записать вывод команды в переменную FILENAME, применив замену команд. Хитро, но запутанно! Мы также проверяем статус завершения программы (строки 7–11). Если пользователь выбирает 'Cancel [Отмена]', чтобы закрыть диалог, *Whiptail* возвращает ненулевой статус завершения, и мы просто оставляем нашу задачу, завершая скрипт. Затем (строка 14) проверяем, принадлежит ли введенное имя папке. Если да, показываем окно «да/нет» (строки 16–23), с вопросом пользователю: менять ли права доступа только для папки или и всех файлов внутри? Здесь мы проверяем статус завершения напрямую и устанавливаем переменную RECURSIVE, если ответ — «да». Строки 26–31 — список галочек, чтобы пользователь задавал любое сочетание прав доступа на чтение, запись и выполнение. Вывод *Whiptail* для них будет строкой вроде "r" "w"

Очищаем эту строку от внешних кавычек и пробелов, передаем ее *tr* (строка 31) и записываем результат в MODES. Финал — строки 38 и 39 выводят готовую команду *chmod* в окне сообщения. »

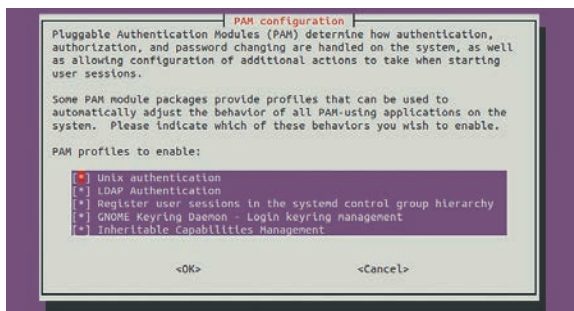
Типы диалоговых окон в Whiptail

Тип диалогового окна	Описание
yesno	Отображает вопрос с кнопками Yes [Да] и No [Нет].
msgbox	Отображает сообщение (с кнопкой ОК для закрытия окна).
infobox	Отображает сообщение и сразу же закрывается.
inputbox	Отображает вопрос и получает от пользователя строку данных в свободном формате. Эта строка затем записывается в <i>stderr</i> .
passwordbox	Аналогично <i>inputbox</i> , но текст не выводится.
textbox	Отображает содержимое текстового файла в диалоговом окне.
menu	Отображает набор вариантов, из которых пользователь выбирает один. В <i>stderr</i> записывается тэг выбранного элемента.
checkboxlist	Отображает набор галочек, но позволяет выбирать любое их сочетание. В <i>stderr</i> записываются тэги выбранных элементов.
radiolist	Элемент управления, аналогичный <i>menu</i> , в котором возможен выбор только одного варианта.
gauge	Отображает индикатор хода процесса, который обновляется на основе значений, считанных из <i>stdin</i> .

OpenLDAP, часть 2

Месяц назад мы настроили хранение учетных записей пользователей в каталоге LDAP. Теперь настроим аутентификацию клиентов для этого каталога.

► Утилита *pam-auth-update* изменяет настройки PAM в зависимости от выбранных источников аутентификации.



В прошлом месяце я начал с краткого описания того, что представляет собой каталог LDAP. Затем мы настроили сервер openLDAP и заполнили каталог, по минимуму, добавив в него группу sales и двух пользователей jane и mary. Мы рассмотрели некоторые пакеты с полезными утилитами LDAP. Первым из них был *ldap-utils*, содержащий полный набор утилит для создания, управления и получения содержимого каталога. Эти утилиты могут работать с любыми данными в LDAP, а не только с учетными записями пользователей. Затем мы установили пакет *ldapscripts*, который содержит более удобные команды для управления учетными записями в LDAP.

Теперь пора обратиться к клиентской части — к тому, как настроить доступ (и аутентификацию) к учетным записям в LDAP с клиентского компьютера.

Прежде всего нужно установить несколько клиентских утилит. В Ubuntu 14.04 я установил всего один пакет:

```
$ sudo apt-get install libnss-ldap
```

Для разрешения зависимостей также были установлены пакеты *authclient-config*, *ldap-auth-client*, *ldap-auth-config* и *libpamldap*. Для одного из этих пакетов, *ldap-auth-config*, нужно задать кое-какие настройки, и установщик запустит *debconf*, запросив у вас эти настройки. Вот мои настройки (разумеется, вам придется изменить некоторые из них в соответствии со своей системой):

```
LDAP server URI: ldap://localhost:389
Distinguished name of the search base: dc=example,dc=com
LDAP version to use: 3
Make local root database admin: Yes
Does the ldap database require login?: No
```

```
LDAP account for root: cn=admin,dc=example,dc=com
LDAP root account password: whateveryouwant
```

Установив все это, нужно настроить две вещи: во-первых, файл «переключения сервиса имен [name service switch]» (*/etc/nsswitch.conf*), чтобы процедуры разрешения имен, такие как *getpwnam()*, обращались к LDAP, и во-вторых PAM, чтобы PAM использовал для аутентификации библиотеку *libpam-ldap*. Есть две программки, которые, как утверждается, могут легко это сделать без ручного изменения файлов настройки, но честно говоря, не знаю, проще ли их освоить или всё-таки отредактировать файлы вручную.

Первая программка — *auth-client config*: ее основная задача — производить изменения в *nsswitch.conf*. Эти изменения задаются в «профилях», определяемых файлами в каталоге */etc/auth-clientconfig/profile.d*. Каждый профиль определяет конфигурацию для определенного сценария. Нам нужен профиль *lac-ldap*. Он применяется так:

```
$ sudo auth-client-config -t nss -p lac_ldap
```

Эта команда изменяет три строки в файле *nsswitch.conf* следующим образом:

```
passwd: files ldap
group: files ldap
shadow: files ldap
```

После этих изменений информация о пользователях и группах должна быть доступна для поиска как в локальных файлах в */etc*, так и в LDAP. Проверить работу функций разрешения из командной строки можно так:

```
$ getent passwd chris mary
chris:x:1000:1000:Chris Brown,,:/home/chris:/bin/bash
mary:x:10000:5000:Mary Brown:/home/mary:/bin/bash
```

Здесь запись для пользователя chris берется из файла */etc/passwd*, но запись для пользователя mary берется из LDAP — это одна из учетных записей, которые мы добавили в прошлом месяце. Аналогично можно получить информацию о группах:

```
$ getent group hackers sales
hackers:x:1004:
sales:*:5000:
```

Опять же, hackers берутся из */etc/group*, а sales — из LDAP.

Настройка PAM

Пока всё хорошо. Теперь пора изменить настройки PAM. Просто скомандауйте

```
$ sudo pam-auth-update
```

Вас попросят выбрать механизм аутентификации, который вы хотите использовать. (См. рис.) Просто выберите 'Unix Authentication' и 'LDAP Authentication', и *pam-auth-update* обновит четыре файла ключей в каталоге */etc/pam.d*: *common-auth*, *common-session*, *common-account* и *common-password*, включив *pam_ldap* в стеки PAM. Эти четыре файла включаются в стеки PAM почти всех приложений, которые знают о PAM.

Может быть, мы уже установили пароль для jane в прошлом месяце, но это было давно, и я понятия не имею, каким он был, так что установим его сейчас. Сложный способ это сделать — воспользоваться программой *ldappasswd* из пакета *ldap-utils*:

```
$ ldappasswd -x -D cn=admin,dc=example,dc=com -W -S
«uid=jane,ou=People,dc=example,dc=com»
New password:
```

sssd

У входа в систему через учетную запись, хранимую на сервере LDAP, есть очевидный недостаток: если вы не сможете достучаться до сервера, то не сможете войти в систему! Это известная проблема офисных работников, желающих пользоваться ноутбуками за пределами офиса, и она может привести к необходимости завести отдельную локальную учетную запись, которой можно пользоваться «оффлайн». Здесь поможет демон *sssd* (system security services — системные сервисы

безопасности). Этот демон появился как проект в Fedora и является неотъемлемой частью RHEL7, но доступен в репозиториях и других (не производных от RedHat) дистрибутивов. Его задача — предоставить общий фреймворк для доступа к серверным хранилищам учетных записей, и он умеет кэшировать логины и пароли, чтобы учетная запись сохранялась при входе в корпоративную сеть и при удаленном подключении. Подробности — на <https://fedorahosted.org/sss>.


```
Re-enter new password:
Enter LDAP Password:
Эти три запроса пароля сбивают с толку. Первые два — это пароль jane, третий — пароль 'root', соответствующей пользователю cn=admin,dc=example,dc=com. Мы задали этот пароль при установке сервера в прошлом месяце. Однако проще воспользоваться скриптом ldapsetpasswd (из пакета ldapscripts), так как он возьмет большую часть необходимой информации из файлов /etc/ldapscripts/ldapscripts.conf и /etc/ldapscripts/ldascript.passwd:
```

```
$ sudo ldapsetpasswd jane
Changing password for user uid=jane,ou=People,dc=example,dc=com
New Password:
Retype New Password:
Successfully set password for user uid=jane,ou=People,dc=example,dc=com
```

Для просмотра только что установленного пароля (точнее, его свертки) можно с помощью поиска по **uid=jane** и вывода атрибута **userPassword**:

```
$ ldapsearch -x -LLL -b dc=example,dc=com -D cn=admin,dc=example,dc=com -W 'uid=jane' userPassword
Enter LDAP Password:
dn: uid=jane,ou=People,dc=example,dc=com
userPassword:: e1NTSEF9N3VKRWVlaWs0
U1BFc1Y1K0NtdjFQK2NNY2ICeTZNMXA=
```

Момент истины

Итак, попробуем войти как пользователь *jane*. Можно либо переключиться в консоль (например, клавишами **Ctrl+Alt+F2**) или выполнить логин с помощью поиска «обратной петли» SSH:

```
$ ssh jane@localhost
jane@localhost's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-39-generic x86_64)
Could not chdir to home directory /home/jane: No such file or directory
```

Ну, мы почти на месте — мы аутентифицируемся через LDAP и правильно определили пользователя:

```
$ id
uid=10001(jane) gid=5000(sales) groups=5000(sales)
```

Проблему с отсутствием домашнего каталога можно решить, создав его вручную, но есть модуль PAM (*pam_mkhomedir*), который сделает эту работу. Чтобы включить этот модуль, можно изменить одну строку в файле */etc/pam.d/common-session*, но не самим — пусть это сделает *pamauth-update*. Создадим новый профиль */usr/share/pam-configs/my_mkhomedir* со следующим содержанием:

```
Name: activate mkhomedir
Default: yes
Priority: 900
Session-Type: Additional
Session:
required pam_mkhomedir.so umask=0022 skel=/etc/skel
```

Теперь примените его, снова запустив *pam-auth-update*. После этого в файл *common-session* будет добавлена следующая строка:

```
session required pam_mkhomedir.so umask=0022 skel=/etc/skel
```

Она велит PAM вызвать модуль *pam_mkhomedir* в начале сеанса. Сделав это, снова попробуйте зайти как пользователь *jane*, и вы должны обнаружить, что ее домашний каталог создан и заполнен файлами из */etc/skel*. Магия!

Снова обратим свое внимание к серверу. Мы рассмотрим три способа добавления пользователя в каталог: путем создания LDIF-файла вручную (очень нудно); с помощью *ldapadduser* (гораздо проще); и с помощью графической утилиты просмотра

debconf

Debconf — механизм сбора от пользователя информации о конфигурации в дистрибутивах на базе Debian. Обычно он запускается автоматически при первой установке пакета, но при необходимости перенастройки пакета его можно запустить и позже, командой **dpkg-reconfigure**. *Debconf* поддерживает

несколько клиентов для отправки запросов и чтения ответов: *dialog* (с текстовой графикой, работает в текстовом терминале), *readline* (простой диалог «вопрос/ответ» в командной строке) и *gnome* (графический). Установите пакет *debconf-doc*, затем наберите **man 7 debconf**, чтобы узнать подробности.

LDAP, такой как LAT (мило, но неудобно, если пользователей несколько). Если вы переходите на LDAP с локальных баз данных, таких как */etc/passwd* и */etc/hosts*, гораздо удобнее установить утилиты для миграции от PADL Software. Они есть в репозиториях Ubuntu, так что установка сложностей не создаст:

```
$ sudo apt-get install migrationtools
```

Эти утилиты по сути представляют собой набор скриптов на Perl (в */usr/share/migrationtools*) с файлом настройки */etc/migrationtools/migrate_common.ph*, также написанным на Perl. Названия этих утилит, например, такие — *migrate_hosts.pl*, которая переносит файл */etc/hosts* в LDAP, и *migrate_passwd.pl*, которая — да! — переносит файл */etc/passwd*. Есть аналогичные скрипты для переноса */etc/aliases*, */etc/group*, */etc/network*, */etc/services* и некоторых других файлов.

Эти утилиты не заполняют каталог напрямую: вместо этого они генерируют файлы LDIF, которые вы затем добавляете в каталог через *ldapadd*. Как мы видели в части 1, LDIF — текстовый формат для «внешнего» представления содержимого каталога LDAP. Также в прошлом месяце мы воспользовались *ldapadd* для минимального заполнения каталога из созданного файла LDIF с узлами **ou=People** и **ou=Groups** и с одной группой (*sales*) и одним пользователем (*mary*).

Надо настроить некоторые записи в файле */etc/migrationtools/migrate_common.ph*. Разумеется, нужно добавить **\$DEFAULT_BASE = «dc=example,dc=com»;** и в моем случае понадобилось изменить RDN для групп с 'ou=Group' на 'ou=Groups':

Теперь, когда набор инструментов настроен, запустите скрипт *migrate_base.pl*, чтобы создать LDIF-файл со всеми записями верхнего уровня, такими как *ou=People,dc=example,dc=com*. В нашем случае часть этой структуры верхнего уровня уже там есть, поэтому наши шаги будут выглядеть так:

```
$ PATH=$PATH:/usr/share/migrationtools
$ migrate_base.pl > base.ldif
```

... измените *base.ldif*, удалив оттуда записи верхнего уровня, которые уже существуют...

```
$ ldapadd -x -D cn=admin,dc=example,dc=com -W -f base.ldif
```

Затем перенесем данные в файл */etc/services*, таким образом:

```
$ migrate_services.pl /etc/services > services.ldif
$ ldapadd -x -D cn=admin,dc=example,dc=com -W -f services.ldif
```

Теперь (на клиенте) измените файл *nsswitch.conf*, чтобы запросы отправлялись (только) в LDAP:

```
services: ldap
```

А мы проявим смелость и переименуем локальный файл сервисов в копию, убрав его с дороги:

```
$ sudo mv /etc/services /etc/services.original
```

Работает? Ну, поищем номер порта сервиса *daytime*:

```
$ getent services daytime
daytime 13/tcp
```

Отлично! Аналогично можно изменить все классические файлы функций-разрешителей: */etc/hosts*, */etc/passwd*, */etc/groups*, */etc/protocols* и т.д. **LXF**

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

Доступно
в AppStore



А если у вас Android, подпишитесь
на Linux Format через Zinio!

 zinio™
Доступно в Google Play



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Больше компьютеров мелких, хороших и разных

Опасайтесь программистов, носящих с собой отвертки!
Леонард Брендвайн

Собралась как-то группа из четырёх инженеров и двух маркетологов, и решили они сделать компьютер величиной с SD-карточку на базе Qualcomm Atheros AR9331. Нечто среднее между Arduino и Raspberry Pi и по возможностям, и по размерам, с Linux в виде OpenWRT внутри и полностью открытыми исходниками, в том числе и на плату. Выставили это предложение на обзор публике посредством Kickstarter и собрали требуемые на проект деньги.

Ничего необычного — типовая современная краудфандинговая история, только вот имена этой группы: Дмитрий, Валерий, Александр, Алексей, Анастасия и Олег. Как минимум кое-кто из них точно живёт в России, а также Гонконге и Новой Зеландии. Мелочь, а приятно, что есть надежда на развитие через энтузиастов.

Называется этот проект Black Swift, и после его запуска планируются продажи внутри России по вполне себе щадящим ценам, что сейчас весьма кстати. Не знаю, что выйдет, но в любом случае — удачи.

PS Harebrained Schemes LLC запустили проект-продолжение вселенной *Shadowrun*, теперь в Гонконге. На момент этой заметки до окончания кампании ещё 12 дней, но уже собрано более 800 тысяч долларов, в 8 раз выше изначальных запросов. Похоже, возник долгоиграющий сериал с поддержкой от его поклонников.
E.M.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Засекать вторжения 68

Шон Конвэй предъявляет Интернету якобы беззащитный порт, а сам из-за угла подсматривает, кто на него покусится.



Оценивать работу 70

Зонды **Михалиса Цукалоса**, понатыканные в разных точках системы, могут составить представление о её производительности. Причем сами много ресурсов не отберут.



Ускорять загрузку 74

С подачи *Gummiboot* **Нейлом Ботвиком** принято решение отправить *Grub* на заслуженный отдых. И это правильное решение: а нечего конфликтовать с UEFI.



Служить на почте 76

Кент Ельчук предпочитает не доверять злокозненному Интернету свою интимную переписку, почему и развернул себе собственный сервер *Squirrel Mail*.



Заклинать змея 80

Змей — сами знаете какой: Python. Но совсем ма-аленький, то есть *Micro*. **Расселу Барнсу**, впрочем, стало ясно, что он способен обогнать и больших.



Расширять память 84

Отдав дань миганию светодиодов, **Михаил Остапкевич** и **Евгений Балдин** замахнулись на задачи более сложные... и сразу же уперлись в нехватку ОЗУ.

АКАДЕМИЯ КОДИНГА



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

Java 88

Знакомьтесь: движок *Matter.js* на *JavaScript* от **Лайама Браммита** — легковесный, но вместивший массу технологий игровой физики.

Erlang 92

Наш сериал про этот язык несколько выдохся, и **Михалис Цукалос** задумал освежить ваши знания. Чем больше практики, тем эффективнее код!

Безопасность: Тестируемся на вторжения

Шон Д. Конвэй показывает, как оставить безопасный сервер слушать порт, а затем открыть его всему миру, чтобы узнать, «кто за вами следит»...



Наш эксперт

Шон Конвэй — вечный студент, чей опыт работы простирается от электронных ламп до комплектарных металл-оксидных полупроводников.



Скорая помощь

Справку по портам разных сервисов можно получить в файле `/etc/services`. Имейте в виду, что это применимо к UNIX и не сильно поможет вам в других ОС вроде Windows.

Главный факт: когда порт компьютера доступен из Интернета, кто-то наблюдает. Мы покажем, как в этом убедиться, и поможем понять необходимость сетевой безопасности, устройств контроля доступа, таких как брандмауэры, и защиты компьютера регулярной установкой обновлений. Мы также покажем, как защитить компьютерный порт, настроить домашний роутер для форвардинга порта в Интернет, а затем поищем в логах посторонние IP, которые к нам подключались.

На рисунке вы видите модем, или резидентный шлюз (residential gateway, RG), предоставляющий доступ к трём узлам, которые показаны справа. Шлюз имеет встроенный хаб/свитч и Wi-Fi, чтобы обеспечивать соединения в локальной сети (LAN). Он использует DHCP, протокол динамического управления хостами, для раздачи частных IP-адресов устройствам, которые подключаются к шлюзу со стороны LAN.

DHCP — это протокол, отвечающий за раздачу и управление IP-адресами, присвоенными узлам в LAN. Диапазоны адресов, которые использует DHCP, берутся из одного из трёх пулов частных IP. Эти адреса закреплены за LAN-стороной шлюза, и их нельзя брать из Интернета.

По левую сторону от шлюза находится глобальная сеть (WAN) — сеть провайдеров, предоставляющих доступ к Интернету. Чтобы шлюз смог работать с WAN, он получает IP-адрес у сервис-провайдера (ISP), который отвечает за управление ими. У ISP есть собственный DHCP, чтобы распространять среди устройств своих клиентов также и внешние IP.

Преобразование сетевых адресов (NAT) — это механизм отсылки IP-трафика из LAN в Интернет. С помощью NAT запросы к Интернету от IP, присвоенных внутри LAN, преобразовываются в запросы от внешнего IP, присвоенного шлюзу, и отсылаются в Интернет. Статическая таблица в шлюзе сохраняет отправленные запросы, поэтому при получении ответа из Интернета

он может быть преобразован из внешнего IP во внутренний и отправлен на устройство в LAN.

Чтобы определить внутренний IP-адрес, присвоенный вашему компьютеру, откройте терминал и введите следующее:

```
sudo ifconfig
```

Для привлечения к себе внимания злоумышленников нужно создать точку входа для запросов подключений. В данном упражнении мы откроем для доступа из Сети стандартный порт SSH: порт 22. SSH — это сетевой протокол, предоставляющий безопасную связь между компьютерами в небезопасной сети. Ответственные за безопасность сетевые администраторы используют SSH для входа на удалённые компьютеры.

Чтобы убедиться, что SSH установлен на UNIX-компьютере, запросите расположение программы командой

```
whereis ssh
```

Если в ответ вы увидите путь `/usr/bin/ssh`, то сервис доступен. Если команда не находит местоположения SSH, установите его через свой менеджер пакетов. На RedHat/Fedora Linux введите

```
sudo yum install ssh
```

На вариациях Debian, например, Ubuntu, введите

```
sudo apt-get install ssh
```

Будучи доступным из Интернета, наш SSH-порт должен быть защищён, чтобы предотвратить взлом системы во время тестирования. Для соединения со службой SSH требуются логин и пароль. Первейший кандидат на взлом — пользователь `root`. Откройте в текстовом редакторе файл `/etc/ssh/sshd_config` и найдите строку с текстом `PermitRootLogin`. Убедитесь, что она содержит только следующее: `PermitRootLogin no`. Сохраните файл и закройте его.

На заключительном шаге настройки позаботимся, чтобы открытый порт был защищён от попыток входа. Создайте в `/etc/hosts.deny` такую запись: `ALL:ALL`.

Место для имён

Вывод команды `cat /etc/passwd` показывает все учётные записи пользователей в системе. Некоторые из них связаны с установленными сервисами. Личностям, пытающимся дорваться до вашей системы, нужны имя пользователя и пароль.

Применение общепотребительных имен для служб, приложений и баз данных, находящихся в `/etc/passwd`, уже избавляет от половины работы по разгадке уравнения с логином/паролем. Администратору важно знать, какие приложения

установлены на компьютере, и изменить стандартные пароли после установки. Пример приложения со стандартными именем пользователя и паролем — `MariaDB`. Если умолчания у СУБД не сменить, злодеи получают всё необходимое для входа.

Журналирование служб

Сервис SSH ведёт журнал (лог) со служебной информацией. Он содержит подробности о том, когда сервис запускался, и записи как об успешных, так и о неуспешных подключениях.

Журналирование — одна из областей, в которых Linux за последнее время подвергся некоторым изменениям. К существующим системам логов *syslog*

и *rsyslog* добавился новый паренёк под названием *journald*.

Место, где хранится журнал, зависит от операционной системы и системы журналирования. Если вы используете один из старых журналов, то вы увидите файл с именем **secure** где-то в **/var/log/**. Его можно просмотреть с помощью текстового

редактора. Если ваша система журналируется с помощью *journald*, то следует использовать инструмент *journal tool* для чтения двоичных файлов журнала.

Подробности о журналировании с *journald* можно найти в статье «Journald: Логи для Systemd» (LXF191).

TCP Wrappers — сервис, который ограничивает доступ к сети определёнными ресурсами, используя файлы **/etc/hosts.deny** и **/etc/hosts.allow**. *TCP Wrappers* служил для защиты сервисов задолго до персональных брандмауэров. **ALL:ALL** запрещает все соединения всем пользователям. Для сравнения, **sshd:ALL** запретило бы всем пользователям SSH-сервисы.

Если вы подумаете, что после этой настройки строка **PermitRootLogin no** в конфигурации SSH стала лишней, то будете правы. *TCP Wrappers* утратил популярность, и некоторые дистрибутивы Linux его уже не содержат. Если вам не требуется тонкая настройка, предоставляемая *TCP Wrappers*, можете его не использовать.

После настройки SSH можно запустить демон следующей командой: на вариациях RedHat Linux —

```
sudo service sshd start
```

а на вариациях Ubuntu Linux —

```
sudo service ssh start
```

В порядке быстрой проверки, что SSH запущен, убедитесь, что порт 22 принимает запросы:

```
netstat -an | more
```

Команда **netstat** с флагом **-an** запрашивает информацию о сетевом соединении без обращения к DNS. Запустив команду с флагом **n**, вы получите имя сервиса, назначенного на этот порт (из файла **/etc/service**).

Проверьте вывод команды **netstat -an** на присутствие одной из следующих строк:

```
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN
```

или

```
tcp 0 0 0.0.0.0:ssh 0.0.0.0:* LISTEN
```

Если порт слушает, попробуйте обратиться к узлу по внутреннему IP, о котором мы говорили в начале:

```
ssh root@<внутренний IP-адрес>
```

При правильной настройке SSH для запрета входа **root** и/или *TCP Wrapper* для безопасности, как описано выше, это не должно сработать. И мы можем быть спокойны насчет безопасности.

Теперь давайте откроем порт Интернету. Это делается с помощью проброса [forwarding] — механизма, с помощью которого шлюзы принимают запрос на порт со стороны WAN и посылают определённому узлу в LAN. Как видно на рисунке, нам нужно настроить модем для переброта любого SSH-подключения к порту 22 из Интернета на наш узел. Как это сделать, зависит от вашего модема и его версии ПО, поэтому вам придётся почитать документацию к нему и разобраться, что делать. Обычно тут возникает необходимость **root**-доступа к шлюзу.

Завершив настройку и компьютера, и шлюза, оставьте форвардинг работающим на несколько

часов и затем следуйте нашей дальнейшей инструкции, чтобы узнать, привлекло ли это какое-либо внимание. Мы ограничимся просмотром файла **/var/log/secure**.

Вы можете отыскать подобные данные и в других системах журналирования, с помощью соответствующих команд (см. врезку вверху).

Проверяем журналы

Когда совершается попытка получить доступ через открытый порт, в логе появится запись от SSH, содержащая данные об IP-адресе посетителя. Просто просмотрите файл текстовым редактором и найдите эту информацию. Затем можно зайти на <https://www.arin.net> и ввести внешний IP того, кто хотел к вам зайти, в поле **Search Whois** — так вы узнаете провайдера, ответственного за выдачу данного IP.

По нашему личному опыту, вы запросто можете обнаружить попытки подключения из всех уголков мира. Не требуется глубокой судебной экспертизы, чтобы догадаться, что это: скоординированная атака или же просто заражённые компьютеры, принадлежащие неудачливым домашним пользователям.

Заметьте, установка на нашем рисунке содержит брандмауэр, но от него здесь мало толку. Как мы увидели, шлюз форвардит входящие соединения (из WAN в LAN) только если они зафиксированы администратором в конфигурации — то есть если вы явно разрешили их проброс. Брандмауэр вступает в игру, если вы хотите ограничиться исходящими сервисами. Если шлюз работает в режиме пропуска PPPoE, то брандмауэр становится точкой, где контролируется доступ к узлу.

Это простое упражнение должно было продемонстрировать вам, что порт, доступный из Сети, привлекает посетителей, даже если вы его не афишируете. Это одна из причин заранее подумать о безопасности и только потом соединиться с Интернетом. Кто-нибудь за вами да следит! LXF

Скорая помощь

Если устройство в LAN имеет внешний IP-адрес, значит, шлюз настроен в режиме пропускания PPPoE. DHCP не назначает внутреннее IP-адреса, а подключённому компьютеру назначается внешний IP-адрес шлюза. При такой конфигурации необходимость в брандмауэре выше, чем при какой-либо другой.

Блок-схема модема/шлюза.



» Подпишитесь на печатную и электронную версии на www.linuxformat.ru/subscribe!

DTrace: Гид по применению

Михалис Цукалос объясняет всё необходимое, чтобы применить удобную утилиту DTrace для проверки производительности вашей системы Linux.



Наш эксперт

Михалис Цукалос

Любит писать и поучать людей насчёт производительности систем, администрирования Unix, программирования и статистики.

DTrace — это инструмент для анализа и отладки ПО. Имеется версия DTrace для Linux, разработанная и поддерживаемая Oracle. Версия Oracle «требует» Oracle Linux, который находится в свободном доступе — его можно скачать с <https://edelivery.oracle.com/linux>, но чтобы использовать DTrace, вам придётся что-нибудь купить у Oracle, так что эта версия не вполне бесплатная. Альтернативная версия DTrace, портированная на Linux, есть на <https://github.com/dtrace4linux/linux> — она свободна совершенно; ею мы и займёмся на нашем уроке.

Два главных козыря DTrace перед аналогичными утилитами — те, что DTrace по своей природе безопасен для вашей деятельности и не забирает много ресурсов компьютера; но вам придётся потратить некоторое время, если вы действительно хотите овладеть им. Будет полезно изучить «внутренности» Linux, чтобы анализировать производительность должным образом, и в конце концов понадобится разобраться в структуре ядра Linux. Ключевым моментом является полное понимание метрики, применяемой для проверки производительности, потому что неправильные метрики приводят к ошибочным заключениям.

Довольно теории; лучше приступим к работе с DTrace и для начала обучимся её установке.

Установка DTrace

На системе с Ubuntu Linux вы можете скачать и установить DTrace, выполнив следующие команды:

```
$ wget ftp://crisp.publicvm.com/pub/release/website/dtrace/dtrace-20140915.tar.bz2
$ bzip2 -d dtrace-20140915.tar.bz2
$ tar xvf dtrace-20140915.tar
$ cd dtrace-20140915/
$ sudo ./tools/get-deps.pl
$ make all
```

Команда `./tools/get-deps.pl` необходима, чтобы автоматически установить все необходимые для Ubuntu пакеты — уж поверьте мне, `get-deps.pl` здорово сэкономит ваше время! Затем надо загрузить модуль DTrace, требующий прав root — это делается так:

```
$ sudo make load
tools/load.pl
12:46:41 Syncing...[Синхронизация...]
12:46:41 Loading [Загрузка]: build-3.13.0-27-generic/driver/dtracedrv.ko
12:46:42 Preparing symbols... [Подготовка символов]
12:46:42 Probes available [Доступно зондов]: 363293
12:46:46 Time [Время]: 5s
```

Команда `sudo make install` установит все необходимые DTrace бинарники в `/usr/sbin`. Убедимся, что модуль DTrace запущен — выполним следующую команду:

```
$ ps -ax | grep dtrace | grep -v grep
2185 ? S< 0:00 [dtrace_taskq]
Чтобы отключить модуль, командуйте sudo /sbin/rmmod dtracedrv.
Для запуска DTrace нужны права root, но определить используемую версию DTrace может и обычный пользователь, командой $/usr/sbin/dtrace -V
dtrace: Sun D 1.9
```

Кроме того, я попытался установить DTrace на систему Debian 7, но процесс прекратился со следующими сообщениями об ошибках:

```
make[2]: *** No targets specified and no makefile found [Не указана цель и makefile не найден].
Stop.
make[1]: *** [kernel] Error 2
tools/bug.sh
make: *** [all] Error 1
```

Вряд ли DTrace нельзя поставить на Debian 7, но если вы только приступаете к изучению DTrace, я бы предложил вам не мучиться и попытаться вместо этого поставить его на Ubuntu Linux. Другие официально поддерживаемые системы Linux — Fedora (`./tools/get-deps-fedora.sh`) и Arch Linux (`./tools/get-deps-arch.sh`).

Простейшее применение DTrace

DTrace (расшифровывается как 'Dynamic Tracing [Динамическое слежение]') обеспечивает способ прикрепить «зонд [probe]» к работающей системе, заглянуть внутрь неё и обнаружить, что она делает. При запуске программы на языке D её текст компилируется в байт-код, проверенный системой безопасности, и затем выполняется в ядре в безопасной виртуальной среде.

Запуская команду DTrace, вы обычно предоставляете информацию о том, что хотите проверить, если только вы не укажете опцию `-l` (это буква l, как в слове 'list'), которая используется просто для вывода списка совпадающих зондов (определённого другими опциями) без возвращения любых данных о реальной производительности.

Опция `-p` указывает имя зонда для отслеживания, а опция `-P` — список таких имен. Команды DTrace допускают несколько параметров `-P` и `-p`. Формат зонда может быть любым из следующих четырёх: `provider:module:function:name`, `module:function:name`, `function:name` или просто `name`.

Опция `-p`, за которой следует ID работающего процесса, захватывает этот ID и кэширует символические таблицы процесса. Опция `-p` может быть несколько.

Использование DTrace с опцией `-c` и путём к программе заставит DTrace запустить программу и начать её отслеживание. К сожалению, на момент написания опция `-c` не была реализована;



Скорая помощь

Если вы администратор различных машин Unix, изучите DTrace непременно: это сильно облегчит вашу жизнь. Если вы администратор только систем Linux, это всё равно хороший выбор.

Зачем создали DTrace?

Хотя отлаживающие утилиты, такие как *strace* и *truss*, могут отслеживать системные вызовы, производимые каким-то процессом, они слишком медленны и потому не подходят для решения проблем с производительностью. Кроме того, ни одна из них не умеет работать на общесистемном уровне, что иногда необходимо.

Вставка в ПО печати сообщений об отладке и других видов сообщений имеет свою цену. Она

невелика, если вы делаете это один раз, но становится огромной, если печать производится всё то время, что вы пытаетесь исправить ошибку или проблему с производительностью.

В Sun Microsystems разработали *DTrace* ещё в 2004 г., чтобы дать представление о рабочих процессах, позволяющее настроить и устранить неполадки приложений и самой ОС. *DTrace* помогает увидеть ПО в работе, то есть показывает, что

происходит за кулисами в масштабах системы, без необходимости что-либо изменять или перекомпилировать, а также позволяет работать в уже запущенной системе и наблюдать запущенные программы или процессы сервера динамически без большой нагрузки.

Поддержка языка программирования D, способного записывать различную информацию, делает *DTrace* ещё более полезным.

это подтвердил и Пол Д. Фокс [Paul D. Fox], автор порта *DTrace* на Linux. Следующая команда должна бы вернуть список всех вызовов функций при выполнении команды `/bin/ls`, но вместо этого она виснет, не возвращая ничего:

```
$ sudo dtrace -n 'pid$target:::entry' -c '/bin/ls'
dtrace: description 'pid$target:::entry' matched 16 probes
[нашлось 16 зондов]
parent: waiting for child [родительский процесс ждёт дочернего]
parent: after waitpid pid=1977 status=137f
rd_loadobj_iter
rd_loadobj_iter: /lib/x86_64-linux-gnu/ld-2.19.so
0x7fdc70a35000
proc-stub:rd_event_enable
proc-stub:rd_event_addr addr=(nil)
proc-stub:rd_event_addr addr=(nil)
proc-stub:rd_event_addr addr=(nil)
```

Обходной путь — запустить *DTrace* на другой ОС с целью обнаружения проблемы и применить исправление на Linux. Я знаю, это не идеально; однако работает! Как сказал мне Пол Фокс, есть вероятность того, что к тому моменту, как вы будете читать это, опция `-c` уже исправится, так что посещайте <https://github.com/dtrace4linux/linux> для обновлений.

Наконец, опция `-s` позволяет скомпилировать указанный исходный файл D-программы, что очень полезно для запуска скриптов, написанных на D (подробнее о D см. далее в этой статье). Опция `-e` означает «выйти после компиляции любых запросов». В конечном счёте, все полезные для вас команды *DTrace* следует запускать в виде скриптов, чтобы сэкономить ваше время и позволить вам автоматизировать процесс. Если в команду *DTrace* не включена ни одна из опций `-e` и `-i`, то будет выполнена указанная программа D.

Зонды и провайдеры

Провайдер разделяет соответствующие зонды на подсистемы. Провайдеры являются библиотеками зондов. Самые важные провайдеры — *dtrace*, *syscall*, *proc*, *profile*, *fbt* и *lockstat*. Вы можете узнать количество доступных провайдеров, запустив команду

```
$ sudo dtrace -l | awk {'print $2'} | sort | uniq | wc -l
```

Зонд — это пользовательский набор инструментов, имеющих прямое соединение с интересующими вас объектами внутри ядра. Как правило, зонд привязан к определённому месту в потоке программы. Когда зонд срабатывает, *DTrace* собирает с него данные и отправляет их вам. Команда `dtrace -l` перечисляет все зонды. Самые полезные имена — **entry** и **return**: они указывают пункты входа и возврата соответствующей функции.

Модуль — это модуль ядра, где находится зонд. Следующая команда перечислит все зонды, предоставляемые провайдером *syscall* (см. рис. 1):

```
$ sudo dtrace -l -P syscall
$ sudo dtrace -l -P syscall | wc -l
1323
```

Наблюдение за полным списком зондов — это очень мудрый способ узнать больше о *DTrace*, к тому же это считается хорошей практикой, особенно если вы хотите продуктивно потратить свободное время.

Следующая команда *DTrace* отслеживает системные вызовы `open()`:

```
$ sudo dtrace -n syscall::open:
dtrace: description 'syscall::open:' matched 4 probes
CPU ID FUNCTION:NAME
0 361374 open:entry
0 361375 open:return
0 361374 open:entry
0 361375 open:return
```

Если ваш зонд не работает, вы получите сообщение об ошибке, похожее на это:

```
dtrace: invalid probe specifier syscall::does_not_exist:: probe
description syscall::does_not_exist: does not match any probes
[не существует: не совпало ни с одним зондом]
```

Программирование DTrace на D

D — это язык структурного программирования, похожий на C и Awk; он не имеет ничего общего с <http://dlang.org>. D снижает нагрузку при сборе и предоставлении данных; поэтому он подходит для производственных сред, где вы не хотите лишний раз отягощать систему.



Если вы не хотите изучать *DTrace*, альтернативы есть: можно использовать команду *perf* (она же *perf_events*), которая тоже почти не нагружает систему; это часть ядра Linux. Другой вариант называется *SystemTap*.

```
SRE -- mtsouk@mtsouk-VirtualBox: ~ -- ssh -- 90x45
mtsouk@mtsouk-VirtualBox:~$ sudo dtrace -l -P syscall | head -43
ID PROVIDER MODULE FUNCTION NAME
361370 syscall x64 read entry
361371 syscall x64 read return
361372 syscall x64 write entry
361373 syscall x64 write return
361374 syscall x64 open entry
361375 syscall x64 open return
361376 syscall x64 close entry
361377 syscall x64 close return
361378 syscall x64 stat entry
361379 syscall x64 stat return
361380 syscall x64 fstat entry
361381 syscall x64 fstat return
361382 syscall x64 lstat entry
361383 syscall x64 lstat return
361384 syscall x64 poll entry
361385 syscall x64 poll return
361386 syscall x64 lseek entry
361387 syscall x64 lseek return
361388 syscall x64 mmap entry
361389 syscall x64 mmap return
361390 syscall x64 mprotect entry
361391 syscall x64 mprotect return
361392 syscall x64 munmap entry
361393 syscall x64 munmap return
```

► Рис. 1. Это всего лишь малая часть вывода очень информативной команды `sudo dtrace -l -P syscall`.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Язык D позволяет определять «действие», где пользователь назначает, как поступить, когда желаемый зонд обнаружен. Он имеет встроенные переменные, включая **execname**; это строка, содержащая имя процесса; **uid**, т.е. ID пользователя; и **pid**, ID процесса.

Программа «Hello World!», написанная на D, выглядит следующим образом:

```
BEGIN {
    trace(«Hello World!»);
}
```

Вы можете сохранить код D в файле **helloWorld.d** и запустить его следующим образом:

```
$ sudo dtrace -s helloWorld.d
dtrace: script 'helloWorld.d' matched 1 probe
CPU ID FUNCTION:NAME
6 1 :BEGIN Hello World!
^C
```

Можно также выполнить программу на D как встроенную, используя следующий формат:

```
$ sudo dtrace -n {program}
```

Вы можете сохранять любые команды *Dtrace* в файле и запускать их как обычный скрипт — это очень удобно. Следующий результат показывает, как такое делается:

```
$ cat helloWorld.d
#!/usr/sbin/dtrace -s
BEGIN {
    trace(«Hello World!»);
}
$ chmod 755 helloWorld.d
$ ls -l helloWorld.d
-rwxr-xr-x@ 1 mtsouk staff 59 Nov 11 11:19 helloWorld.d
```

Полезнейшее одиночное действие — **printf**, отображение информации на экране — подобно вызову **printf**, функции C. Проиллюстрируем использование **printf**:

```
$ sudo dtrace -n 'syscall::open:entry { printf(«%s %s», execname, copyinstr(arg0)); }'
CPU ID FUNCTION:NAME
0 361374 open:entry vminfo /var/run/utmp
0 361374 open:entry upowerd /sys/devices/LNXSYSTM:00/device:00/PNP0A03:00/PNP0C0A:00/power_supply/BAT0/present
```

Приведённая команда отслеживает исходный вызов системного вызова **open(2)** и при обнаружении печатает имя процесса и путь, использующий действие **printf()**.

Следующая программа D состоит из трёх частей, прямо как программа AWK:

```
$ cat beginEnd.d
#!/usr/sbin/dtrace -s
BEGIN
{
    printf(«Hello World!\n»);
    printf(«Press Control+C to exit.\n»);
}
syscall::read:entry
{
    printf(«Program %s is asking for %d bytes\n», execname, arg2);
}
END
{
    printf(«Goodbye World!\n»);
}
```

```
$ sudo ./beginEnd.d
dtrace: script './beginEnd.d' matched 4 probes
CPU ID FUNCTION:NAME
0 1 :BEGIN Hello World!
Press Control+C to exit.
0 361370 read:entry Program beginEnd.d is asking for 8192 bytes
0 361370 read:entry Program sshd is asking for 16384 bytes
...
0 361370 read:entry Program sshd is asking for 16384 bytes
0 2 :END Goodbye World!
```

Как видите, провайдер *DTrace* имеет зонд **BEGIN** и зонд **END**. **BEGIN** включается при запуске программы, прежде всех прочих действий, а **END** — в конце программы. Вы можете использовать зонд **BEGIN** для инициализации переменной и печати результата. Зонд **END** очень полезен для печати отчётов и сводок.

Хотя весь доступный код D, который вы найдёте в Интернете, не всегда работает в Linux, вам следует попробовать прочитать и понять его, чтобы лучше изучить *DTrace* и, может быть, изменить его и заставить работать на вашей системе Linux.

Функции статистики

Язык D поддерживает статистически обобщающие [aggregating] функции, помогая создавать полезные сводки вместо показа полного результата *DTrace*. Обобщение — это специальный тип переменной. Поддерживаемые функции обобщения — **avg** (среднее арифметическое), **count** (количество вызовов), **sum** (суммарное значение), **min** (минимальное значение), **max** (максимальное значение), **stddev** (стандартное отклонение), **lquantize** (линейное распределение) и **quantize** (квадрат дисперсии).

Следующая команда печатает общее число системных вызовов для процесса с именем **<process_name>**, и это весьма удобно для обнаружения всяких подробностей о способе работы процесса:

```
$ sudo dtrace -n 'syscall:::entry /execname == «<process_name>» / { @[probefunc] = count(); }'
```

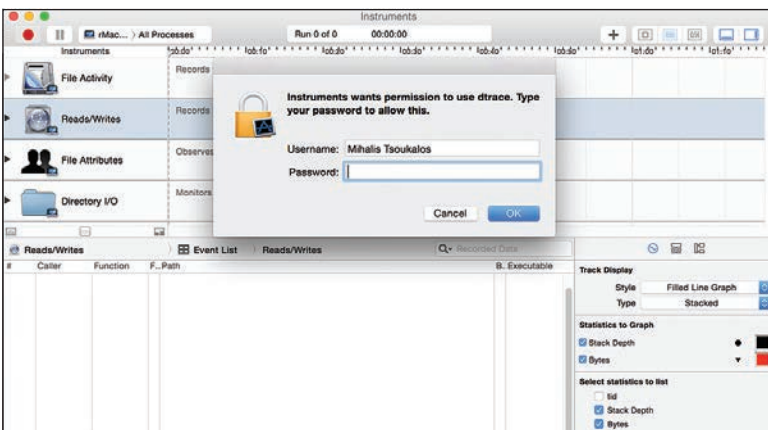
Следующая команда считает все системные вызовы для всех процессов с именем **'sshd'**:

```
$ sudo dtrace -n 'syscall:::entry /execname == "sshd" / { @[probefunc] = count(); }'
dtrace: description 'syscall:::entry ' matched 661 probes
^C
accept 1
chroot 1
...
fstat 152
```



Скорая помощь

Когда вы пытаетесь решить проблему с производительностью, один вопрос тащит за собой другой. Не сдавайтесь; просто продолжайте отвечать на возникающие вопросы!



► Рис. 2. Инструмент *Instruments* из OS X — графическое приложение, получающее информацию через *DTrace*. Надеемся, похожий инструмент появится и для Linux.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

```
open 181
mmap 207
close 219
read 347
```

Следующая команда считает все системные вызовы для процессов с PID 778 (которые уже запущены):

```
$ sudo dtrace -n 'syscall::entry /pid == 778/ { @[probefunc] = count(); }
```

Полезные команды-однострочники

Есть ещё небольшие команды, которые полезно знать. Например, такая команда печатает число системных вызовов на каждую работающую программу:

```
$ sudo dtrace -n 'syscall::entry { @num[execname] = count(); }'
dtrace: description 'syscall::entry ' matched 661 probes
^C
sudo 1
...
sshd 25
dtrace 3233
```

Если ваша система неизвестно почему тормозит, первым делом запустите эту команду, чтобы обнаружить, какая программа может являться причиной проблемы.

А вот команда, которая не только отслеживает все системные вызовы `open()`, но также печатает имя и путь процесса, вызвавшего `open()`:

```
$ sudo dtrace -n 'syscall::open:entry { printf("%s %s", execname, copyinstr(arg0)); }'
dtrace: description 'syscall::open:entry ' matched 2 probes
dtrace: error on enabled probe ID 2 (ID 361374:
syscall:x64:open:entry): invalid address (0x7f9babcb20f8) in
action #2 at DIF offset 28
CPU ID FUNCTION:NAME
0 361374 open:entry vminfo /var/run/utmp
0 361374 open:entry vminfo /var/run/utmp
```

Следующая команда действительно впечатляет, показывая реальную мощь *DTrace*. Она печатает количество прочитанных байтов, сгруппированное по процессам:

```
$ sudo dtrace -n 'syscall::read:return { @[execname] = quantize(arg0); }'
```

Подобным же образом обнаруживается то же для записи:

```
$ sudo dtrace -n 'syscall::write:return { @[execname] = quantize(arg0); }'
```

Можно написать и команду *DTrace* для отслеживания ввода/вывода диска и печати ID процесса, его имени и размера операции ввода/вывода в байтах:

```
$ sudo dtrace -n 'io:::start { printf("%d %s %d", pid, execname, args[0]->b_bcount); }'
dtrace: description 'io:::start ' matched 2 probes
CPU ID FUNCTION:NAME
0 113 :start 1887 head 8192
```

```
0 113 :start 1808 sshd 16384
0 115 :start 1808 sshd 68
...
0 113 :start 1889 tar 392
```

Первая строчка результата показывает, что команда `head` процесса с ID 1887 была задана операцией ввода/вывода с размером 8192 байта. Чтобы посмотреть определённую программу, можно отфильтровать результат предыдущих команд с помощью *grep*:

```
$ sudo dtrace -n 'io:::start { printf("%d %s %d", pid, execname, args[0]->b_bcount); }' | grep -w sshd
```

Или вот команда *DTrace* для подсчёта исходящих соединений, отслеживанием вызовов `connect()`:

```
$ sudo dtrace -n 'syscall::connect:entry { @[execname] = count(); }'
```

Следующая команда аналогично сосчитает входящие соединения, по вызовам `accept()`:

```
$ sudo dtrace -n 'syscall::accept:return { @[execname] = count(); }'
```

Чтения и записи через сокет, которые соответствуют вызовам `read()` и `write()`, сгруппированные по процессу с заданным именем, отследит команда

```
$ sudo dtrace -n 'syscall::read:entry,syscall::write:entry { @[execname] = count(); }'
dtrace: description 'syscall::read:entry,syscall::write:entry '
matched 4 probes
^C
gmain 1
dtrace 3
...
sshd 6
vminfo 55
```

Последняя предлагаемая однострочная команда сосчитает вызовы функции, относящиеся к `ext4`:

```
$ sudo dtrace -n 'fbt::ext4_*:entry { @[probefunc] = count(); }'
dtrace: description 'fbt::ext4_*:entry ' matched 458 probes
^C
ext4_bread 1
ext4_data_block_valid 1
...
ext4_readdir 2
ext4_htree_store_dirent 27
ext4_getattr 29
ext4_has_inline_data 30
```

По мере роста мощности компьютерных систем ПО становится всё более сложным и тоже, в свою очередь, требует решения возникающих с ними проблем. Время на изучение *DTrace* или другого подобного инструмента будет потрачено не зря.

Кроме того, помните, что *DTrace* — это та разновидность инструментов, которую следует изучать на практике, а не в теории, так что приступайте к ней прямо сейчас! **LXF**

Через месяц:
Raik
NoSQL

DTrace против Linux DTrace

DTrace уже доступен на многих системах Unix, включая Solaris, FreeBSD и OS X. Следующий результат от компьютера с Linux и Mac, работающего на OS X 10.10, показывает общее число зондов в каждой системе:

```
(LINUX) $ sudo dtrace -l | wc
363293 1816253 31401331
(MAC OS X) $ sudo dtrace -l | wc
270581 1734358 39349011
```

Как вы поняли, чем больше зондов обеспечивает ваша система Unix, тем лучше!

Возможно, вы спросите, зачем вам вообще беспокоиться о других версиях *DTrace*. Ответ прост: потому что версии для этих платформ «окончательные», так что версиям Linux когда-нибудь нужно будет предложить те же функции. В OS X даже имеется *Instruments*, графический инструмент, использующий *DTrace* (рис. 2).

Помните, что не каждая команда *DTrace*, которую вы сумеете откопать в Интернете, с ходу заработает в вашей системе Linux, потому что подобные команды обычно пишутся с учётом ядра Solaris. Тем не менее, большая их часть должна работать без всяких проблем или с минимальными изменениями.

Основополагающее руководство по *DTrace* см. на <http://www.dtracebook.com>.

Gummiboot: Ускорим загрузку

Нейл Ботвик изучает программу, во многом, очень многим способную заменить *Grub*, если, конечно, у вас более-менее новый компьютер.



Наш эксперт

У Неила Ботвика море опыта по части загрузки, ведь у него в каждой комнате по компьютеру; а вот в перезагрузке, после ухода с Windows на Linux, он не силен.



В прошлом месяце мы разбирались, как настроить внешний вид и параметры загрузчика *Grub 2* по своему вкусу. В этом месяце мы пойдём другим путём и покажем, как обойтись без него вообще. Это стало возможным благодаря UEFI. Хотя по адресу UEFI были кое-какие нарекания, в основном они касались функции Secure Boot и ее способности усложнить жизнь производителям и некоммерческим операционным системам. Однако UEFI — это далеко не только Secure Boot. Одной из главных его особенностей является принципиально иной, по сравнению с загрузчиком, способ запуска ОС. Исторически загрузчик был

необходим для построения таблицы разделов MS-DOS, поскольку ОС запускалась с помощью кода, содержащегося в первом секторе жёсткого диска. В самом этом секторе было всего 512 байт, и там также хранилась информация по таблице (отсюда и лимит в 4 раздела и неуклюжее решение с расширенными и логическими разделами); в общей сложности, этого кода было достаточно, чтобы где-нибудь на диске мог располагаться реальный загрузчик. В прошлом в Linux эту роль выполнял *LILO*, а для систем Microsoft был загрузчик Windows. В UEFI всё иначе: здесь код может загружаться непосредственно из первого раздела жёсткого диска, с тем ограничением, что он форматирован в файловой системе FAT. Это называется EFI System Partition или ESP.

И это означает, что вы можете обойтись совсем без загрузчика, и ядро с соответствующими настройками будет запущено самим UEFI. На практике понадобится хотя бы минимальный менеджер загрузки, чтобы можно было изменять её параметры или переключаться между несколькими системами. Одним из таких менеджеров и является *Gummiboot* (<http://freedesktop.org/wiki/Software/gummiboot>).

Оставим Grub про запас

Поскольку UEFI позволяет запускать сразу по несколько программ, например, ядер ОС, с помощью нажатия клавиши в процессе загрузки, загрузчиков может тоже быть несколько. Как правило, это удобно не столько для повседневного использования, сколько для тестирования. Ведь можно поэкспериментировать с *Gummiboot*, при этом не удаляя *Grub*, на случай, если первая попытка установки *Gummiboot* окажется неудачной.

Скорее всего, *Gummiboot* есть в репозиториях вашего дистрибутива, или же можете рискнуть и скачать последнюю версию с <http://cgkit.freedesktop.org/gummiboot>. Установив *Gummiboot* на свою систему, вам нужно также установить её в ESP вот этой командой:

```
gummiboot install
```

Подразумевается, что ваш ESP располагается в `/boot`. Если нет, используйте опцию `--path`. Необходимо, чтобы все файлы, которые *Gummiboot* использует при загрузке, находились в ESP, поскольку это единственная файловая система, которую ему позволяет прочитать UEFI, так что проще всего использовать ESP как `/boot`. Опция `install` делает EFI исполняемым в `/boot` и позволяет UEFI с него загружаться. Правда, в некоторых случаях последний шаг окажется невозможным из-за внутренних настроек ядра; тогда можно установить *efibootmgr* и добавить *Gummiboot* в меню с помощью команды

```
efibootmgr --create --loader \EFI\gummiboot\gummibootx64.efi --label Gummiboot -v
```

Двойной обратный слэш необходим, поскольку UEFI предполагает описание пути в стиле MS-DOS, а в оболочках Linux слэш



» Меню загрузки *Gummiboot* — никаких прикрас, зато всё очень быстро.

является знаком экранирования следующего символа; поэтому удвоение подчёркивает, что вам нужен именно слэш. Можно выполнить *efibootmgr* и без аргументов, чтобы посмотреть, какая конфигурация UEFI установлена на вашей машине. Вы увидите список опций, их порядок и время ожидания до запуска.

Создаём меню загрузки

Не спешите перезагружаться: нужно ещё сообщить *Gummiboot*, какие системы у вас установлены и как их следует загружать. По сравнению с *Grub2* в *Gummiboot* файлы настройки кажутся до смешного простыми. Главный файл называется **/boot/loader/loader.conf**, и в нём всего два параметра: время ожидания в секундах и имя записи, которая должна загружаться по умолчанию. Можно также приукрасить меню загрузки с помощью *background* и *splash*. Сами записи определяются в индивидуальных файлах в **/boot/loader/entries** и выглядят так:

```
title My Distro
linux /vmlinuz-3.17.2
initrd /initrd-3.17.2
options root=/dev/sda3
```

Имя каждого загрузочного файла должно заканчиваться на **.conf**, а без **.conf** оно используется в разделе по умолчанию **loader.conf**. Вот и все настройки для базовой загрузки *Gummiboot*. Добавить в меню дополнительные строки, например, варианты восстановления системы или загрузки с или без X можно, создав соответствующие файлы **.conf** в **/boot/loader/entries**. Вы также можете указать в файле **.conf** номер версии — это очень удобно, когда у вас стоит несколько версий одного и того же дистрибутива: добавьте **title** с именами каждого из них и **version** с номером соответствующей версии ядра.

Если вам нужна заставка, изображение должно быть в формате BMP и, как и другие файлы, на которые ссылается *Gummiboot*, должно храниться в **/boot**.

В *Gummiboot* нет одной важной функции, которая есть в *Grub*: установить пароль на изменение записей меню. Так что существует риск нарушения защиты в том случае, если физический доступ к машине, например, к ноутбуку, не ограничен. Однако если на нём у вас содержится какая-то важная информация, вы всё равно наверняка используете шифрование файловой системы, и при отсутствии этого пароля изменение записей загрузочного меню взломщику ничего не даст.

Настраиваем UEFI

Другим полезным инструментом настройки загрузки в UEFI является *efibootmgr*. Мы уже прибегали к нему, чтобы велеть системе использовать *Gummiboot*, но с его помощью можно менять и другие параметры основного процесса. При использовании UEFI настройки хранятся в системе в виде переменных; с *efibootmgr* это можно изменить. Чтобы увидеть текущие настройки, выполните такую команду без параметров:

```
efibootmgr
BootCurrent: 0001
```

```
mount - Mounts a file system on a block device
mv - Moves one or more files or directories to another location
openinfo - Displays the protocols and agents associated with a handle
pause - Prints a message and waits for keyboard input
pci - Displays PCI device list or PCI function configuration space
ping - Ping a target machine with UEFI network stack
Press ENTER to continue, 'q' to exit:
reconnect - Reconnects one or more EFI drivers to a device
reset - Resets the system
rm - Deletes one or more files or directories
sermode - Sets serial port attributes
set - Displays or modifies EFI Shell environment variables
shift - Shifts batch file input parameter positions
smbiosview - Displays SMBIOS information
stall - Stalls the processor for the specified number of microseconds
telnetmgmt - Change terminal type
time - Displays or changes the current system time
timezone - Displays or sets time zone information
touch - Updates filename timestamp with current system date and time
type - Displays file contents
unload - Unloads a EFI driver
ver - Displays EFI Firmware version information
vol - Displays or changes a file system volume label
```

```
Timeout: 3 seconds
BootOrder: 0001,0002,0000,0003
Boot0000* GRUB2
Boot0001* Linux Boot Manager
Boot0002* UEFI: Built-in EFI Shell
Boot0003* Hard Drive
```

Как мы видим, установлено два менеджера загрузки, *Grub2* и 'Linux Boot Manager', который мы только что загрузили с последнего (BootCurrent: 0001), и время ожидания составляет 3 секунды. 'Linux Boot Manager' — это имя, которое даёт себе *Gummiboot* после установки. Его вы не сможете поменять ни через *Gummiboot*, ни через *efibootmgr*, но сможете создать новую запись командой **efibootmgr --create**, упомянутой выше. Загрузившись и проверив, что всё работает, можете удалить первую запись:

```
efibootmgr --bootnum 0001 --delete-bootnum
```

В ходе тестирования *Gummiboot* разумно использовать *Grub* как загрузчик по умолчанию и определить порядок загрузки, расположив номера записей в нужной последовательности.

```
efibootmgr --bootorder 0000,0004,0002,0003
```

Теперь надо нажать ту клавишу, с помощью которой в вашей системе вызывается меню загрузки, и выбрать вариант *Gummiboot* или использовать вот этот трюк:

```
efibootmgr --bootnext 0004
```

Тогда указанное действие будет выполнено по умолчанию только при следующей загрузке, а в дальнейшем порядок будет прежним.

Удовлетворившись тем, как работает *Gummiboot*, можете определить этот порядок загрузки по умолчанию. Вероятно, вам также захочется изменить время ожидания в *efibootmgr*, чтобы не просиживать впустую. Кроме того, некоторые из этих опций можно настроить прямо в меню UEFI, смотря какая у вас система. На моём настольном ПК эта возможность очень пригодилась, а на ноутбуке страница настроек UEFI порадовала своей простотой! **LXF**

➤ Возможно, вам никогда не придётся использовать EFI Shell, но если что, там масса различных команд.

Скорая помощь

Gummiboot умеет работать только с устройствами UEFI. Для систем с более ранними версиями BIOS в духе MS-DOS он не подойдёт. Но на большинстве компьютеров последних двух лет UEFI быть должен.

UEFI Shell

Возможно, вы заметили, что в списке параметров *efibootmgr* значатся таковые для EFI Shell. Это простейшая программа-оболочка, с помощью которой можно просматривать и изменять настройки, а также запускать приложения UEFI. Это аналог командной оболочки *Grub*. Набор её функций зависит от версии системы. Например, в спецификации UEFI 2.0 есть хорошая альтернатива *efibootmgr*,

под названием *bfcg*, тогда как в более ранних версиях её нет. UEFI — штука довольно новая, так что обновления для вашей материнской платы могут заметно улучшить его возможности.

Список команд, доступных для вашей системы, вместе с их кратким описанием, вы можете посмотреть с помощью

```
help -b
```

Вывод каждой из них можно постранично останавливать, используя **-b** — так же, как в Linux команду можно шаг за шагом просматривать с помощью **less**. Эта оболочка очень низкоуровневая, так что лучше не лезть сюда, если вы не слишком уверены в своих действиях или у вас что-то не получилось с *efibootmgr*. Хотя команды для просмотра информации о системе вполне безопасны.

SquirrelMail: Начнем, пожалуй

Кент Ельчук показывает, как создать собственный почтовый сервер с нуля, установив и настроив гибкий *SquirrelMail*.



Наш эксперт

Кент Ельчук — web-разработчик, который рьяно кодирует web-приложения, пишет посты в блог и книги, а также много экспериментирует с виртуальными машинами и Raspberry Pi.

Скорая помощь



Если ваш провайдер не разрешает почтовому серверу отправлять почту через порт 25, его всегда можно попросить разрешить это, или отправлять исходящую почту через такого провайдера, как Gmail.

Желание установить собственный почтовый сервер может возникнуть по разным причинам — скажем, ради повышенной безопасности или лучшего контроля. Мы покажем, как настроить *SquirrelMail* так, чтобы отправлять и получать почту из браузера в любой точке мира.

Если у вас установлен LAMP, можете сразу переходить к разделу «Почтовый сервер» ниже. Если нет, установка *Apache*, PHP и *MySQL* займет всего пару минут:

```
sudo apt-get install apache2
sudo apt-get install mysql-server libapache2-mod-auth-mysql
php5-mysql
sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

Почтовый сервер

Чтобы начать работать с почтой, нужно установить почтовый сервер. Есть несколько вариантов серверов, включая *Exim*, но мы воспользуемся *Postfix*. Для установки *Postfix* выполните следующую команду:

```
root@Joe-VirtualBox:~# apt-get install postfix postfix-mysql
dovecot-core dovecot-imapd dovecot-lmtpd dovecot-mysql
dovecot-pop3d
```

Выполнив простые инструкции, вы сможете создать самоподписанный сертификат или пойти другим путем и не создавать SSL. Что касается имени хоста, можете оставить вариант по умолчанию — localhost. Если вы планируете использовать доменное имя, укажите его в поле “System Mail Name”. Так, если адрес локального хоста — `foo@example.org`, то доменным именем будет `example.org`.

Следующий шаг после настройки почтового сервера — установка пакета web-почты. *SquirrelMail* — легкий, развивается уже не первый год и часто применяется крупными хостинг-провайдерами. Для его установки выполните следующую команду от имени суперпользователя-root:

```
apt-get install squirrelmail squirrelmail-locales
```

Для настройки *SquirrelMail* можно просто воспользоваться командой `squirrelmail-configure`. Но сначала нужно сделать еще кое-что. Первым делом создадим мягкую ссылку:

```
root@Joe-VirtualBox:~# ln -s /etc/squirrelmail/apache.conf /etc/
apache2/conf-enabled/squirrelmail.conf
```

(Обратите внимание, что всего несколько месяцев назад команда для создания мягкой ссылки на сервер LAMP была бы немного другой. Но времена меняются, и названия файлов и каталогов в различных пакетах могут немного переделываться. Скажем, вчерашний каталог `conf.d` сегодня называется `conf-enabled`.)

Затем выполните команду `restart apache2`. Теперь, открыв браузер и набрав в адресной строке `http://localhost/squirrelmail/src/login.php`, вы должны увидеть страницу входа в систему. Хотя на самом деле войти в систему еще нельзя, тот факт, что вы эту страницу видите, означает, что установка прошла корректно.

После этого надо открыть страницу `http://localhost/squirrelmail/src/configtest.php`. Вы увидите ошибку IMAP:

```
ERROR: Error connecting to IMAP server "localhost:143".
Server error: (111) Connection refused
```

Не волнуйтесь, эта ошибка случается, и ее легко исправить. Чтобы это сделать, сначала запустите команду

```
apt-get install nmap
```

а затем другую, от имени root, для сканирования портов:

```
nmap localhost
```

Эта команда выведет список портов, и вы сможете посмотреть, не отсутствует ли какой-то порт. Например, если нет порта 143 или 25, нужно исправить ошибки и убедиться, что соответствующие сервисы запущены. С сервером *SquirrelMail* основную проблему обычно представляет *Dovecot*, и перезапустить его можно командой

```
service dovecot reload
```

Еще один удобный способ устранить ошибки IMAP — команда

```
sudo netstat -a | fgrep imap
```

Эта команда должна вернуть строку с “LISTEN”. Если она ее вернула, это означает, что сервер IMAP работает и запущен.

Другая альтернатива — запустить команду `netstat -nl4`. Теперь проверим, что вы можете подключиться к порту 143 с помощью telnet, для чего скопируем

```
telnet localhost 143
```

(Если у вас нет telnet, сначала установите ее командой `apt-get install telnetd`, затем выполните указанную выше команду.)

Если подключение по telnet прошло успешно, вы увидите строку “Connected to Localhost”, и мы сможем двинуться дальше.

Перезагрузите *Dovecot* указанной ранее командой, и после этого вы сможете пользоваться *SquirrelMail*. Если теперь открыть страницу `http://localhost/squirrelmail/src/configtest.php` в браузере, в нижней части страницы должна появиться строка “Congratulations, your SquirrelMail setup looks fine to me! [Поздравляем, с установкой SquirrelMail все отлично!]”

Настройка SquirrelMail

Теперь рискните зайти на `http://localhost/squirrelmail/src/login.php` со своим логином и паролем Ubuntu. Если всё получится, вы должны увидеть страницу `webmail.php` и, вероятно, ошибку с INBOX. Однако другие ссылки — а именно INBOX.Drafts, INBOX.Sent и INBOX.Trash — должны работать правильно.

Если что-то пойдет не так, вы сможете найти ошибки в файле `/var/log/mail.log`. Распространенная ошибка — отсутствие у пользователя файла настроенного файла INBOX в нужном месте. В лог-файле появится сообщение о том, что выбрать каталог со входящей почтой невозможно:

```
Nov 17 12:30:26 Joe-VirtualBox dovecot: imap(Joe): Error:
Opening INBOX failed: Mailbox isn't selectable
```


Использование nmap

С помощью *nmap* удобно сканировать порты своего локального хоста. Но с ее помощью также можно просканировать любой сайт, так как это ценная утилита для анализа безопасности сети.

Если просто запустить *nmap* с нужным хостом (localhost, IP-адрес или URL), она просканирует 1000 портов, чего вполне достаточно для нашего примера.

```
root@Joe-VirtualBox:~# nmap localhost
Starting Nmap 6.40 ( http://nmap.org ) at 2014-11-20
13:35 PST
Nmap scan report for localhost (127.0.0.1) Host is up
(0.000068s latency).
Not shown: 989 closed ports
```

PORT	STATE	SERVICE
22	tcp open	ssh
23	tcp open	telnet
25	tcp open	smtp
80	tcp open	http
110	tcp open	pop3
143	tcp open	imap
631	tcp open	ipp
783	tcp open	spamassassin
993	tcp open	imaps
995	tcp open	pop3s
3306	tcp open	mysql

Не считая нашего упражнения, *nmap* — еще и ценная утилита сканирования портов, с помощью

которой можно узнать всю информацию о хостах в сети, такую как открытые порты, используемые технологии и их версии, операционные системы и маршруты трассировки.

С помощью трех следующих команд можно просканировать порты без обнаружения версии. Вторая команда просканирует только порт 4000, а последняя — диапазон портов.

```
sudo nmap -sV <sitename/ip>
sudo nmap -sV -p 4000 <sitename/ip>
sudo nmap -sV -p 4500-5000 <sitename/ip>
```

Длинный список параметров *nmap* можно найти на странице http://linuxcommand.org/man_pages/nmap1.html.

В этом случае нужно создать файл INBOX в `var/mail/username`. Файл, в котором можно изменить эту настройку — `/etc/dovecot/conf.d/10-mail.conf`, а `mail_location` задает локаль для INBOX. Владелец файла может быть этот пользователь (по умолчанию `usergame`), а правами доступа — `777`. Это не каталог, а пустой файл.

Чтобы *SquirrelMail* заработал, нужно выполнить еще кое-какие изменения. Важно указать расположение почты — оно определяет, где находятся почтовый ящик и папка для входящих писем. Его всегда можно узнать с помощью следующей команды.

```
root# grep -r «mail_location» /etc/dovecot
```

В данном случае, оно указывает на `/etc/dovecot/conf.d/10-mail.conf`.

Два основных файла, в которых нужно произвести изменения — `/etc/postfix/main.cf` и `/etc/dovecot/conf.d/10-mail.conf`.

На первый взгляд всё вроде должно работать нормально. Но если вы попытаетесь отправить письмо, оно вряд ли достигнет своего назначения. Сам сервер работает, но чтобы заработала отправка почты, следует разрешить перенаправление порта 25 на роутере для локального IP. Если IP не перенаправляется, то вы сможете отправлять, но не получать почту. Разрешить перенаправление просто: зайдите в роутер и перенаправьте локальный IP, чтобы принять адрес.

Отныне *SquirrelMail* должен работать. Свое имя и адрес электронной почты можно задать в Options > Personal Information [Параметры > Личная информация].

Далее, откройте терминал и переключитесь на пользователя `root`. Затем добавьте нового пользователя и задайте его пароль.

```
useradd test
passwd test
```

Теперь создайте новый каталог для этого пользователя в домашнем каталоге, а в этом каталоге создайте подкаталог `mail`. Необходимые команды, в том числе для установки владельца и прав доступа, приведены ниже.

```
Mkdir /home/test
mkdir /home/test/mail
chown test:test mail
chmod -R 777 mail
```

Не беспокойтесь о каталогах `INBOX.Drafts`, `INBOX.Sent` и `INBOX.Trash`, о которых мы говорили ранее — они будут созданы автоматически. На этом этапе письма должны отправляться от одного пользователя другому.

Отправка почты на другие серверы

Отправка почты другому пользователю в пределах вашего доменного имени — это одно, отправка почты через сеть — совсем другое. Основной камень преткновения обычно — старый добрый

порт 25. Он может блокироваться провайдером; в этом случае провайдер может помочь с решением проблемы. Иногда он даже может разблокировать этот порт для вас. Проверить, заблокирован ли порт, можно с помощью `Telnet`. В худшем случае ваш IP-адрес может оказаться в черном списке, что значительно затруднит настройку сервера.

На данный момент у вас есть два варианта отправки почты. Первый — отправлять ее с почтового сервера, второй — воспользоваться для отправки промежуточным сервисом, таким как `Gmail`. Последний вариант удобен в том случае, если ваш IP в черном списке или провайдер блокирует порт 25.

Чтобы немного упростить жизнь, есть два файла `main.cf` (один для промежуточного сервиса и один без), которые подойдут для любого варианта. Для каждого из файлов нужно изменить `myhostname` и `myorigin`, указав в них свое имя домена вместо `example.org`. Также нужно добавить IP-адрес своей сети в параметр `mynetworks`. В файлах указан адрес по умолчанию `192.168.0.109`, в конце строки с предшествующим пробелом.

Если порт 25 не заблокирован провайдером, вы сможете отправлять и получать почту без промежуточного сервиса. Тогда можно воспользоваться фрагментом кода из файла настройки `Postfix /etc/postfix/main.cf`, приведенным ниже. Все комментарии удалены, чтобы фрагмент не был слишком большим.

```
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no
append_dot_mydomain = no
readme_directory = no
```

DYNAMIC DNS

The DDNS feature allows you to host a server (Web, FTP, Game Server, etc...) using a domain name that you have purchased (www.whateveryournameis.com) with your dynamically assigned IP address. Most broadband Internet Service Providers assign dynamic (changing) IP addresses. Using a DDNS service provider, your friends can enter your host name to connect to your game server no matter what your IP address is.

Sign up for D-Link's Free DDNS service at www.dlinkddns.com.

Save Settings Don't Save Settings

DYNAMIC DNS SETTINGS

Enable Dynamic DNS :

Server Address : <<
Select Dynamic DNS Server

Host Name :

Username or Key :

Password or Key :

Verify Password or Key :

Timeout : (hours)

Status : **Disconnected**



Скорая помощь

MX Toolbox — фантастический ресурс для дешифровки длинных почтовых заголовков и их проверки по спам-листам. Вы найдете его на <http://mxtoolbox.com/EmailHeaders.aspx>.



» В некоторых роутерах есть бесплатный сервис DNS, и вы можете использовать свой домашний IP-адрес для хостинга или почтового сервера.

```
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = example.org
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = example.org
mydestination = example.org, Joe-VirtualBox, localhost, localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.109
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

➤ Команда `nmap localhost` сообщит вам важнейшую информацию об используемых портах. Вам понадобятся порты 25 и 143.

Если вы можете отправлять письма только на адреса своего нового сервера, можно настроить отправку писем через промежуточный сервер с помощью SMTP. Для этого надо выполнить три следующих изменения в файле `/etc/postfix/main.cf` — просто добавьте эти строки в конец файла:

```
relayhost = [smtp.gmail.com]:587
smtp_use_tls=yes
```

```
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd.
```

Если вы новичок в таких делах, полную закомментированную версию кода можно найти в файле `/usr/share/postfix/main.cf.dist`. (Заодно взгляните в файл `/usr/share/doc/postfix/TLS_README.gz` в пакете `postfix-doc`, ради информации о включении SSL в клиенте SMTP.)

Выполнив изменения в файлах `main.cf` и `/etc/postfix/sasl_passwd`, запустите следующую команду:

```
sudo postmap /etc/postfix/sasl_passwd
```

Команда необходима для того, чтобы `Postfix` знал, как найти файл с именем пользователя и пароля Gmail. Код в файле `sasl_passwd` будет выглядеть как приведенная ниже строка, с тем исключением, что нужно будет указать настоящий пароль своей учетной записи Gmail.

```
[smtp.gmail.com]:587 example@gmail.com:
my_secret_password_here
```

Теперь перезагрузите `Postfix`, и всё готово.

```
sudo service postfix reload
```

Отныне при отправке письма ваш домашний сервер воспользуется вашей учетной записью Gmail и аутентифицируется там, и письмо в конце концов будет доставлено получателю. (Но учтите, что входящие сообщения не будут пропущены через Gmail или любой другой промежуточный сервис: они попадут в ваш почтовый ящик напрямую от отправителя.)

При отправке письма через промежуточный сервис заголовки писем будут содержать информацию и об исходном отправителе, и об учетной записи Gmail. Удобный способ преобразования длинных заголовков — на <http://mxttoolbox.com/EmailHeaders.aspx>.

В поле `Return-Path` получатель увидит, что письмо пришло с сервиса Gmail, но так отправителя не скрывать: после анализа заголовков с помощью MX Toolbox или вручную при проверке почты вы сможете отследить обратный путь и определите, что адрес, с которого пришло письмо (в данном случае) — Joe@example.org: заголовок `X-Google-Original-From` говорит, что Google получил письмо с этого адреса. Поля `Received` также раскроют информацию об организациях и серверах, которые обрабатывали сообщение в процессе передачи, и будут содержать IP-адрес почтового сервера, с которого сообщение было отправлено.

Вложения, DNS и SSH

Теперь вы можете успешно получать и отправлять почту и, возможно, хотите обновить свой `SquirrelMail`, сделав так, чтобы он был способен отправлять и принимать вложения.

```
root@kent-VirtualBox #mail: ~
Reading state information... Done
nmap is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 235 not upgraded.
root@kent-VirtualBox
#mail:~# service dovecot restart
stop: Unknown instance:
dovecot start/running, process 3305
root@kent-VirtualBox
#mail:~# nmap localhost

Starting Nmap 6.40 ( http://nmap.org ) at 2014-11-19 15:55 PST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000073s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
```

Использование почтового клиента

Хотя почта через web-интерфейс удобна и доступна из любой точки мира, у вас может возникнуть желание воспользоваться для доступа к почте любимым почтовым клиентом, таким как `Thunderbird Mail`.

По умолчанию, `Thunderbird Mail` настроит вашу учетную запись с IMAP. Поэтому у вас всегда будет копия старых и новых писем. Однако для получения почты через порт 110 можно воспользоваться протоколом POP и настроить перенаправление портов на роутере.

Вложения получены

По умолчанию, вложение будет сохраняться во фрагменте текста, закодированном в `base64`, и храниться во входящих письмах каталога `/var/mail/username`. Если вы отправите себе вложение, дойдите до конца файла и прочтите его. Чтобы

перейти в конец файла в редакторе `Vi`, наберите `G` и нажмите `Enter`. Если вы хотите сохранить свои вложения, переместите их в каталог `/var/lib/squirrelmail/attachments` или просто загрузите на свой компьютер.

В блоке ниже показана информация о полученном вложении:

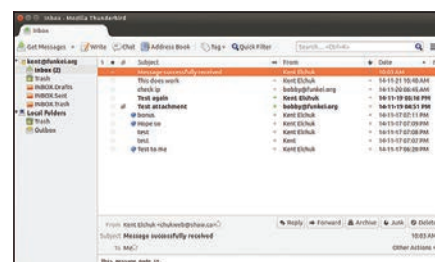
```
Content-Type: application/vnd.oasis.opendocument.text;
name=myfilename.odt
Content-Disposition: attachment;
filename=myfilename.odt
Content-Transfer-Encoding: base64
```

Если по какой-то причине вы захотели опустошить свой файл `inbox`, это делается командой

```
sudo truncate -s 0 /var/mail/username
```

Как насчет отправленных вложений? Опять же, как и для полученных вложений, каждое из них

будет закодировано `base64` и будет храниться в файле `/home/username/INBOX.Sent`. Чтобы увидеть это самим, отправьте письмо с приложенным файлом. Затем откройте файл `/home/username/INBOX.Sent` в редакторе вроде `Vi` и перейдите в конец файла. Там вы увидите вложение, закодированное в `base64`.



Это очень просто. Для начала зайдите в каталог `/var/lib/squirrelmail`. Там можно создать каталог для вложений с правами доступа 777 и с владельцем и группой `www-data`. Вот простые команды для этого:

```
root:/var/lib/squirrelmail# mkdir attachments
root:/var/lib/squirrelmail# chmod -R 777 attachments
root:/var/lib/squirrelmail# chown www-data:www-data attachments
```

Если у вас Raspberry Pi, флэшка, раздел с Linux или ПК, на котором установлен только Linux, можете установить статический IP, чтобы ничего не приходилось менять каждый раз при перезагрузке компьютера или сбросе роутера. Давайте посмотрим, как заставить это работать.

Ваш новый почтовый сервис будет доступен локально по адресу `localhost/squirrelmail/src/login.php`. Если вы хотите, чтобы он был доступен через Интернет, то обратиться к нему можно с любого компьютера по ссылке <http://51.68.8.248/squirrelmail/src/login.php>. (Конечно, если вы планируете использовать IP-адрес своего провайдера, вместо 51.68.8.248 надо подставить его.)

Желая использовать доменное имя, надо будет выполнить еще несколько шагов, после чего *SquirrelMail* будет доступен по ссылке вроде <http://example.org/squirrelmail/src/login.php>. Есть множество бесплатных сервисов DNS, на которых можно связать доменное имя с IP-адресом. На самом деле, такой бесплатный сервис может быть даже в роутере. В дополнение к бесплатному сервису DNS можно установить пакет под названием *Bind*. Однако это скорее всего будет излишним, если только у вас нет нескольких статических IP-адресов от провайдера и хороших идей по управлению ими. А это бывает дорого и для маленького сайта невозможно. Так что можно ограничиться бесплатным сервисом DNS, связав доменное имя с IP-адресом домашнего компьютера или задав его с VPS или выделенного сервера.

Далее все зависит от того, для какого компьютера с Linux в домашней сети настроено перенаправление портов в роутере. Любой компьютер, где настроено перенаправление порта 80, автоматически станет основным компьютером для доменного имени.

Чтобы сохранить на компьютере с Linux статический IP-адрес, можете открыть файл `/etc/network/interfaces` и задать статический IP. Можно даже пойти еще дальше, зайдя в панель управления роутером и зарезервировав IP-адрес для этого устройства.

В простом коде ниже изменены строки 2 и 3, а строки после третьей — добавлены и привязывают IP-адрес сети к этому компьютеру. После загрузки системы с этими изменениями компьютер получит локальный сетевой адрес 192.168.0.109. Шлюз `[gateway]` — это IP-адрес роутера.

```
# interfaces(5) file used by ifup(8) and ifdown(8)
#auto lo
#iface lo inet loopback
auto lo eth0
iface lo inet loopback
iface eth0 inet static
address 192.168.0.109
netmask 255.255.255.0
gateway 192.168.0.1
```

Чтобы изменения вступили в силу, скомаундите

```
/etc/init.d/networking restart
```

Теперь когда все готово и работает, вы можете захотеть дополнительно установить сервер `ssh`, чтобы можно было подключаться к системе и управлять ей из любой точки мира, где есть интернет-соединение. Сервер устанавливается простой командой

```
sudo apt-get install openssh-server
```

Чтобы это работало, обязательно включите перенаправление портов для этого IP-адреса в роутере. Для всех приведенных примеров этот IP-адрес — 192.168.0.109. В конце концов, может оказаться, что перенаправить нужно все порты, указанные в статье. Если у вас есть постоянный доступ по SSH, то вы можете работать с любыми файлами где и когда вам удобно. Тем, кто работает удаленно, жизнь без этого представить почти невозможно.

Борьба со спамом

Это почти всё, что необходимо для полнофункционального почтового сервера, позволяющего получать и отправлять почту. Однако в современном мире есть еще одна вещь, которую нужно принять в расчет — это спам. Хотя существуют пакеты для фильтрации спама, которые можно установить, например, *SpamAssassin*, можно попробовать не усложнять дело и отфильтровать массу спам-сообщений с помощью *Postfix*.

Полезный файл, имеющийся в *Postfix* — `/usr/share/postfix/main.cf.dist`. С помощью этого файла можно выбирать различные параметры и получить справку по ним. Один фантастический параметр — `smtpd_recipient_restrictions`, который может даже проверять письма по спам-спискам. Очень хорошая web-страница, которая окажет неоценимую помощь — <http://www.postfix.org/postconf.5.html>.

Как описано на этой странице, можно создать собственный белый список, черный список и даже список пользователей, которые смогут отправлять письма с вашего сервера. Файл настройки, который нужно изменить — `/etc/postfix/main.cf`. Другой вариант — попробовать *SpamAssassin*, как описано ниже. **LXF**



Воспользовавшись статическим IP-адресом и зарезервировав его в роутере, вы сэкономите массу времени по сравнению с изменением всех локальных адресов в сети. Если ваш IP-адрес изменился, нужно изменить перенаправление портов в роутере и изменить файлы настройки, указав в них новый IP-адрес.

Использование SpamAssassin

SpamAssassin — бесплатный пакет, который можно установить для фильтрации спама. Его с такими почтовыми серверами, как *Postfix* или *Exim*, используют многие компании, предоставляющие услуги хостинга.

Ниже приводится список команд, которые позволят установить и включить *SpamAssassin* и сделать так, чтобы он работал с *Postfix*.

Руководство по установке очень краткое, так как многие параметры заданы в значениях по умолчанию. Если вы захотите углубиться в настройку, то потенциальных параметров и пояснений хватит на целую книгу.

```
sudo -s
apt-get install spamassassin spamc
groupadd spamd
useradd -g spamd spamd
```

```
mkdir /var/log/spamd
chown spamd:spamd /var/log/spamd
vi /etc/default/spamassassin
```

В этом файле замените строку

```
ENABLED=0
```

на строку

```
ENABLED=1
```

Многие параметры можно задать в файле

```
/etc/default/spamassassin. Как вы видите в ниже-
следующей команде, к конфигурации был добавлен
-s /var/log/spamd/spamd.log для записи в файл
spamd.log.
```

```
OPTIONS="--create-prefs --max-children 5 --helper-
home-dir -s /var/log/spamd/spamd.log"
```

Полный список параметров можно отыскать на сайте <http://spamassassin.apache.org/full/3.0.x/dist/doc/spamd.html>.

Для запуска сервиса *SpamAssassin* выполните команду

```
sudo service spamassassin start
```

Она позаботится о *SpamAssassin*. Теперь нужно внести небольшое изменение в файл `/etc/postfix/master.cf`, как показано ниже.

Как видите, строка, начинающаяся с `submission`, раскомментирована, а новая строка с `-o content_filter=spamassassin` добавлена с пробелом слева. Если не добавить пробел перед этой строкой, после перезагрузки *Postfix* появится сообщение об ошибке.

```
submission inet n - - - smtpd
-o content_filter=spamassassin
#-o syslog_name=postfix/submission
```

Проделав эти изменения, можно запустить команду `service postfix reload`, и на этом всё.

Micro Python: А ну, приступим

Рассел Барнс овладевает Micro Python, чтобы превратить маленький, но мощный микроконтроллер Pyboard в маленькую, но мощную USB-мышь.



Наш эксперт

Рассел Барнс пишет о компьютерах и технологиях почти 20 лет. У него также есть сайт и подкаст, посвященные Raspberry Pi, www.raspi.today.

Micro Python и микроконтроллер Pyboard были успешно реализованы с помощью средств, собранных на Kickstarter в конце 2013 г. (<http://kck.st/17T8bW6>). Идея состояла в том, чтобы сделать Python достаточно компактным и эффективным для работы на микроконтроллерах. Зачем? В Python масса замечательных вещей, но он явно не приспособлен для операций с критичным временем выполнения. Достаточно будет сказать, что Google использует Python для массы вещей — но не для тормозов своего самоуправляемого автомобиля.

Ужатию ядра Python и изменение (среди прочего) способа работы компилятора позволили сделать Micro Python быстрее и эффективнее любых версий Python. При компиляции на полноценном ПК это штука революционная, и скоро ее могут взять на вооружение большие коммерческие проекты.

С точки зрения удобства и производительности микроконтроллер Pyboard лежит где-то между Raspberry Pi и Arduino. Pi гораздо мощнее среднего микроконтроллера, но немного проигрывает в задачах, в которых критично время (то же торможение автомобиля). В противоположность Pi, Arduino довольно медленный, но выигрывает, когда необходимо управлять несколькими входами и выходами с высокой точностью.

С быстрым процессором Cortex M4, 32-битным аппаратным модулем для обработки чисел с плавающей точкой, аппаратным генератором случайных чисел и почти всеми на свете видами коммуникационных интерфейсов (включая I2C) у Pyboard есть не только мозги, но и мускулы. К тому же, если учесть, что программируется она на уже знакомом вам Python, воспользоваться Pyboard впечатляюще просто.

На плате — 30 входов/выходов общего назначения [GPIO] (в том числе 14 аналоговых входов с 12-битными АЦП и два аналоговых выхода), датчик температуры, часы реального времени и трехосный акселерометр, но при этом она весит всего около 6 г и имеет размеры 33×40 мм. Она достаточно мала для использования в переносных устройствах и подойдет для воплощения почти всех мыслимых идей. Неудивительно, что этот контроллер подороже Arduino — £24 без расходов на доставку — но он стоит каждого потраченного гроша.

Подключение

Микроконтроллер Pyboard и язык Micro Python связаны гораздо более интересными отношениями по сравнению с другими подобными платами. В отличие от Arduino, вы не просто заполняете крошечный сегмент памяти кодом на «псевдо-С», написанным на компьютере; с помощью Micro Python вы также можете напрямую взаимодействовать со своей платой по последовательной линии связи. Micro Python в роли псевдооперационной системы и его скриптовый язык позволяют быстро и легко протестировать или отладить код (подробнее об этом позже).

Как и Arduino, Pyboard подключается к компьютеру через USB, и таким образом в него можно загрузить новые скрипты — просто подключите его порт MicroUSB к любому свободному USB-порту компьютера (тут подойдет любой кабель для зарядки телефонов на Android). Но, в отличие от Arduino, в Pyboard довольно много памяти, и при подключении к компьютеру контроллер определяется как обычный накопитель. Поэтому вы сможете написать скрипт на Micro Python в любимом текстовом редакторе или среде разработки и просто перетащить его во внутреннюю память устройства или на карту памяти MicroSD. Да, вот так всё просто.

По умолчанию на Pyboard вы найдете четыре файла: файл `readme`, драйвер Windows (для настройки последовательного устройства USB), `boot.py` и `main.py`. Как и в любом приложении Python, `main.py` — основной скрипт, содержащий ваше приложение, но он не выполняется, пока не отработает `boot.py` — скрипт, настраивающий поведение Pyboard при включении. Попозже мы подробно рассмотрим этот аспект Pyboard, чтобы наша плата вела себя как обычное HID-устройство (Human Interface Device — Устройство для взаимодействия с человеком) USB.

Привет, мир!

Но сначала давайте перейдем к основам лицензированного MIT языка Micro Python и контроллера Pyboard, написав простую программу “Hello, World”. На плате Pyboard есть четыре светодиодных индикатора, последний из которых можно плавно погасить с помощью PWM (Pulse Width Modulation — широтно-импульсная модуляция, ШИМ), поэтому давайте попробуем что-нибудь сделать с ним, с целью убедиться, что всё работает правильно.

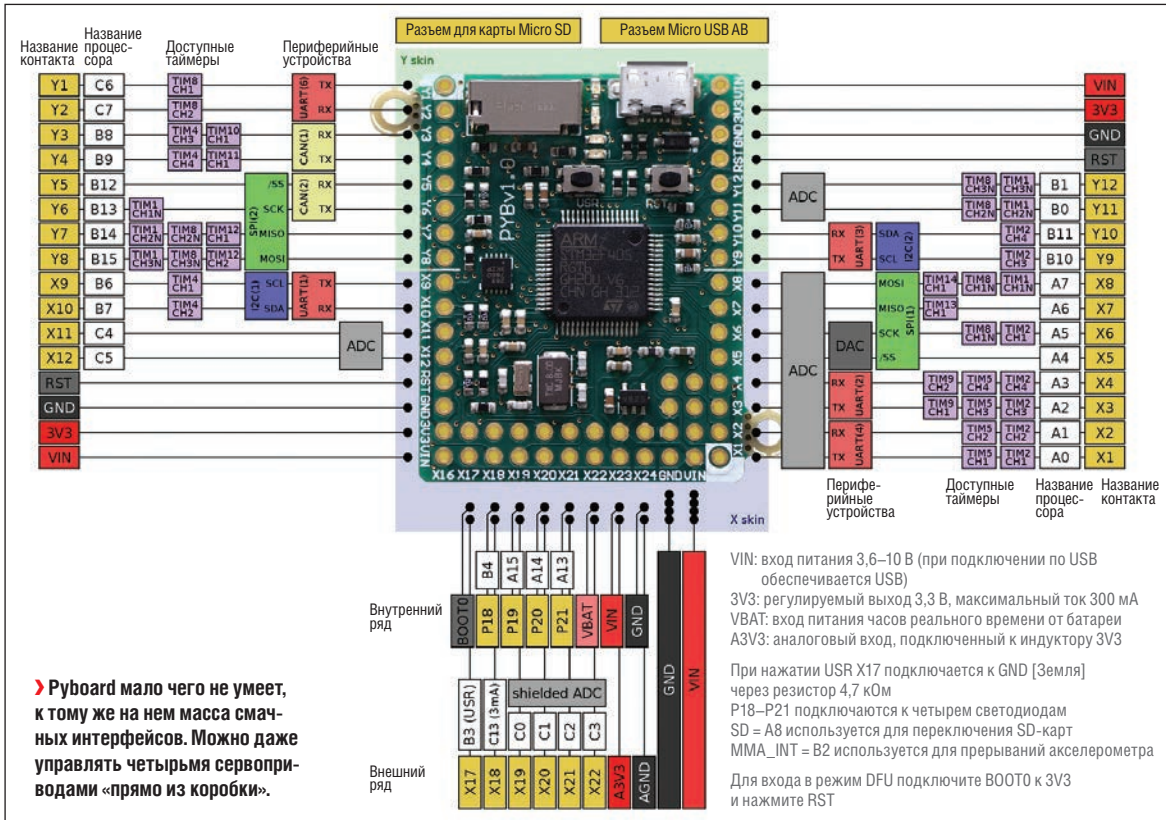
Прежде всего подключите Pyboard к компьютеру кабелем MicroUSB. В большинстве случаев внутренний накопитель устройства смонтируется автоматически, как обыкновенная USB-флешка. Если в вашем дистрибутиве накопителя нужно монтировать вручную, просто наберите `lsblk` в терминале, чтобы увидеть все подключенные диски, и смонтируйте нужный вам, командой `mount /dev/sdbx (sdbx — имя соответствующего устройства)`. Смонтировав устройство, дважды щелкните на значке диска и откройте файл `main.py` в текстовом редакторе.

Это пустой документ с закомментированной строкой, говорящей “put your code here [добавьте код сюда]”. Давайте так и сделаем:

```
led = pyb.LED(4)
brightness = 0
while True:
    if brightness < 255:
        brightness += 5
        led.intensity(brightness)
        pyb.delay(50)
    else:
```



С «голым железом» быстро и удобно общаться через последовательное соединение. В `Screen` открыть командную строку Python можно командой `screen /dev/ttyAMC0`.



Скорая помощь

Лучший итог совместимости Pyboard с картами MicroSD не в том, что между ними можно быстро и легко переключаться для запуска скриптов Micro Python, а в том, что их можно использовать для журналирования данных.

brightness = 0

Как вы, наверное, поняли, этот простой скрипт представляет собой бесконечный цикл, в котором один из четырех светодиодов Pyboard переключается между минимальной и максимальной яркостью. В первой строке мы создаем объект **led**, который относится к светодиоду, поддерживающему ШИМ. Во второй строке мы создаем переменную **brightness**, а в третьей начинаем бесконечный цикл, в котором снова и снова изменяем яркость светодиода. Если эта яркость не максимальная, мы просто пошагово (в данном случае, на 5 единиц) увеличиваем яркость светодиода. Затем мы устанавливаем соответствующую интенсивность светодиода и выполняем задержку в 50 мс, чтобы увидеть постепенное изменение яркости светодиода.

Чтобы скрипт заработал, сохраните измененный файл **main.py**, извлеките (размонтируйте) Pyboard и нажмите кнопку RET (сброс) на самом Pyboard. Эта кнопка находится прямо под портом MicroUSB, рядом с кнопкой USR (пользовательская), программируемой нефиксируемой кнопкой, которую мы включим в скрипт чуть позже.

Если все хорошо, то голубой светодиод, расположенный между портом MicroUSB и картой MicroSD, должен замигать. Если он не мигает, это либо ошибка в коде, либо не работает Pyboard — в этом случае попробуйте пообщаться с Pyboard напрямую и посмотреть, что тут может быть...

Python OS

Пока мы пользовались Pyboard и Micro Python почти так же, как и любым другим микроконтроллером: подключили его к компьютеру, написали какой-то код, загрузили его в контроллер и — надеюсь — полюбовались симпатичным мигающим светодиодом. Даже из того, что мы уже сделали, очевидно, что Pyboard удобнее в использовании, чем большинство микроконтроллеров, так как для его работы не нужны специализированная среда разработки,

проприетарные драйверы или программы; а во всем остальном он ведет себя как любой другой популярный контроллер.

Давайте всё изменим, подключившись к Pyboard напрямую по последовательному подключению и обнаружив настоящую командную строку. Фактически это интерактивная среда программирования Python, но не только: по сути, это операционная система для Pyboard — из нее можно управлять всеми параметрами контроллера (и даже файловой системой)!

Для подключения к интерактивной среде Pyboard можно воспользоваться любой из популярных программ для последовательного подключения, но мы выберем *Screen*. Как можно узнать из файла **readme** на Pyboard, для подключения к контроллеру нужно набрать следующую команду в окне терминала на компьютере с Linux:

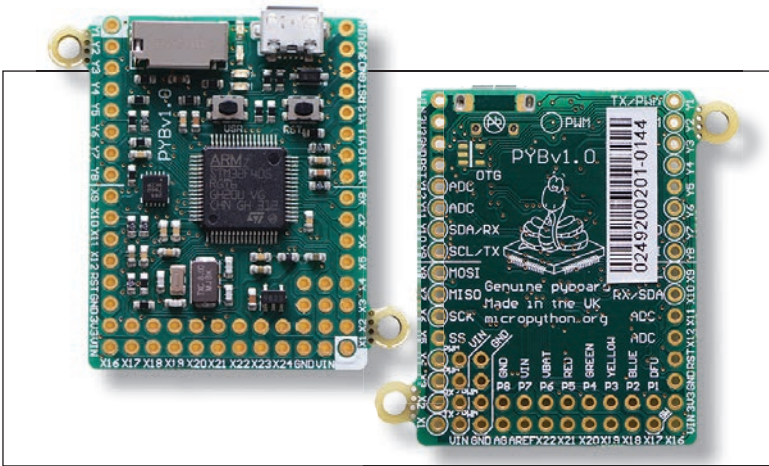
```
screen /dev/ttyAMC0
```

Для пользователей Mac все очень похоже (**screen /dev/tty.usbmodem***), но для пользователей Windows — как и можно было подумать — все совсем иначе. Им нужно зайти в Диспетчер устройств, щелкнуть правой кнопкой мыши на «неизвестном устройстве» и обновить драйвер, указав файл **pybcdc.inf** с контроллера. Для доступа к интерактивной командной строке также понадобится программа вроде *PuTTY*. Если вам по какой-либо причине приходится пользоваться Windows, полные инструкции можно найти на <http://bit.ly/windowsrepl>.

Поговори со мной на Python

Если открывается черный экран, нажмите Enter для запуска интерактивной командной строки Python; но если после этого всё равно ничего не появилось, убедитесь, что у вас не запущен скрипт на Pyboard. Это никак не описано в документации, но если скрипт выполняется, то сразу ничего не откроется. Простейший способ это исправить — удалить код из **main.py** и перезапустить Pyboard вручную кнопкой RET (сброс).

» Не хотите пропустить номер? Подпишитесь на www.linuxformat.ru/subscribe/!



➤ Несмотря на малый по сравнению с большинством плат Arduino размер (всего 33 × 40 мм) Pyboard может похвастаться мощнейшим 32-битным процессором Cortex.

Когда интерактивная командная строка откроется, вы сможете — как и ожидалось — просто «говорить со своим контроллером на Python». Если у вас раньше были проблемы со скриптами, то вы обнаружите, что отлаживать их в интерактивной командной строке вдесятеро быстрее. Чтобы убедиться в этом, наберите такие строки, для проверки нажимая Enter после каждой строки:

```
led = pyb.LED(1)
led.on()
led.off()
```

Красный светодиод должен загораться и снова гаснуть после каждой команды. Из командной строки REPL доступны все веселые штуки в Pyboard. С помощью библиотеки `pyb` можно считать состояние акселерометра, измерить температуру или получить параметры других подключенных периферийных устройств, например, сервоприводов. Также можно выполнить мягкий сброс Pyboard, нажав Ctrl+D в командной строке.

С помощью хорошо известной библиотеки OS Python можно продвинуться во взаимодействии с Pyboard гораздо дальше. На сайте <http://bit.ly/REPLos> вы найдете полный список сервисов операционной системы; они довольно просты в использовании и не требуют пояснений.

Например, как и в обычном Python, можно вызвать `os.listdir()` и увидеть, какие файлы доступны в текущем каталоге. Если подключена SD-карта с файлами, то для просмотра файлов на внутреннем накопителе можно воспользоваться `os.listdir('/flash')`, а для файлов на SD-карте — `/sd`. Также можно создавать и удалять каталоги и делать многое другое.

Но пожалуй, самая важная библиотека — та, что предназначена для работы с внутренними устройствами Pyboard — `pyb`.

Создатель Micro Python сделал всё возможное для того, чтобы вы могли максимально воспользоваться «железом», которое уже есть на плате. Например, можно переключать состояния светодиода, просто вызывая `.toggle()`, или пользоваться кнопкой USR как переключателем. В последнем варианте, пока кнопка нажата, вызов ее функции в скрипте вернет True, в противном случае — False. Вот незатейливый пример для начала:

```
light = pyb.LED(4)
button = pyb.Switch()
while True:
    if button():
        light.on()
    else:
        light.off()
```

Помните, что если вы копируете этот код на Pyboard с компьютера, то для запуска кода нужно размонтировать плату и нажать кнопку сброса. Но если этот код набирается в интерактивной командной строке, можно остановить скрипт нажатием Ctrl+C, как и обычный скрипт на Python, выполняющийся в командной строке. Подробный обзор возможностей библиотеки имеется на http://bit.ly/pyb_lib, а здесь ограничимся замечанием, что он выходит далеко-далеко за рамки этой статьи.

Доступ к акселерометру

Окончательная цель этой статьи — превратить Pyboard в псевдомышь, курсор которой можно было бы перемещать, просто наклоняя плату и используя параметры акселерометра. Как можно было предположить, за взаимодействие с акселерометром в Pyboard отвечает отдельный класс библиотеки `pyb`. В командной строке создать объект акселерометра и получить его параметры тоже довольно просто:

```
wobble = pyb.Accel()
while True:
    print("X:", wobble.x(), "Y:", wobble.y(), "Z:", wobble.z())
    pyb.delay(1000)
```

Всё, что мы сделали — создали объект акселерометра и вывели его координаты X, Y и Z, за которыми мы будем наблюдать, двигая плату. Скрипт будет плеваться координатами, пока плата не скончается от износа, поэтому нам придется остановить его нажатием Ctrl+C. Мы внесли задержку в одну секунду, чтобы результаты не обновлялись слишком часто, но отфильтрованный набор координат также можно получить, вызвав в примере выше `wobble.filtered_xyz()`.

Всё это довольно стандартные вещи, но в том и прелесть Micro Python — пользователи могут считать, что пишут просто на Python 3.3. При желании мы могли бы записывать параметры акселерометра на SD-карту для последующего анализа или управлять с их помощью чем-то еще на плате (например, при наклоне платы в разных направлениях можно включать разные светодиоды

Квадрокоптер

По словам разработчика Micro Python и Pyboard, оба были предназначены для того, чтобы создать маленького робота. Дэмиен Джордж [Damien George] с помощью сообщества занимается именно этим — в своем проекте по созданию квадрокоптера на Micro Python. Хотя тот еще на ранних стадиях разработки, в разделе о проектах форума Micro Python <http://forum.micropython.org>, уже есть несколько страниц комментариев и конструктивные схемы проекта.



➤ Из любви к робототехнике Дэмиен Джордж создал Pyboard.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



или — с помощью класса `Servo` — смастерить самонастраивающегося робота).

Мы собираемся считывать значения параметров акселерометра и передавать их в маленький класс под названием `hid`, предназначенный для перемещения курсора мыши. Эта функция принимает кортеж или список, содержащий кнопки и координаты, представляющие события мыши, и выполняет эти события. Как вы вскоре увидите, код довольно простой, но чтобы заставить компьютер считать, что `Ryboard` — это на самом деле мышь, а не накопитель, нужно преодолеть еще несколько препятствий.

Сокровища, скрытые в HID

Как упоминалось ранее, на верхнем уровне файловой системы накопителя `Ryboard` вы найдете файл `boot.py`. Скрипт запускается сразу же после включения `Ryboard` и предназначен для того, чтобы изменять поведение микроконтроллера. Взгляните на содержимое скрипта, и вы увидите, что некоторые параметры закоментированы. Нас интересует последний, который велит `Ryboard` «выступать в качестве последовательного устройства и мыши».

Если раскомментировать последнюю строку, сохранить файл и перезагрузиться, вы не сможете получить доступ к файловой системе на запись, чтобы записать код для управления файловой системой. Поэтому раскомментируем и сохраним `boot.py` и не будем перезагружаться, пока не поместим следующий код в `main.py` и не сохраним его:

```
import pyb
move = pyb.Accel()
button = pyb.Switch()
while True:
    if button():
        pyb.hid((0, move.x(), -move.y(), 0))
        pyb.delay(25)
```

Здесь, как и ранее, мы создали объекты кнопки и акселерометра, а также бесконечный цикл с условием “while True”. Внутри этого цикла мы проверяем, нажата ли кнопка, и перемещаем курсор мыши (с помощью метода `hid`) в координаты, определяемые параметрами X и Y акселерометра. Задержка необходима для стабильной работы — ее легко изменить по своему вкусу.

Сохранив скрипт, можно размонтировать и перезагрузить `Ryboard`. Теперь, если вы нажмете кнопку `USR` и будете наклонять контроллер, курсор мыши начнет перемещаться по экрану. Уже весело; но раз нельзя на чём-нибудь щелкнуть, какая же это мышь? Мы исправим беду, но пока нужно сбросить `Ryboard`...

Безопасный режим

Поскольку мы изменили поведение `Ryboard` при подключении к компьютеру, теперь нельзя просто отключить и потом подключить его заново, чтобы добыть доступ к файловой системе и отредактировать код. Поэтому следует заставить контроллер перезагрузиться в безопасный режим. Для этого нажмите и удерживайте `USR`, а затем нажмите кнопку `RST`. Подождите одну секунду, затем отпустите `USR`, и оранжевый светодиод должен быстро мигнуть несколько раз — это будет означать, что вам всё удалось.

Теперь вы снова должны получить доступ к файловой системе. Чтобы вернуть `Ryboard` в обычный режим, можете убрать комментарий `#` перед последней строкой файла `boot.py`. Если вы хотите, чтобы контроллер остался в режиме мыши, но при этом нужно изменить скрипт, просто проигнорируйте `boot.py` и откройте `main.py`. Чтобы нажатие кнопки на `Ryboard` распознавалось как щелчок мыши, пригодится такой код:

```
import pyb
move = pyb.Accel()
button = pyb.Switch()
while True:
    pyb.hid((button(), move.x(), -move.y(), 0))
    pyb.delay(25)
```

Запустив его, а затем размонтировав и перезагрузив `Ryboard`, вы увидите: при наклоне контроллера курсор мыши автоматически перемещается, а нажатие кнопки `USR` имитирует щелчок мыши на экране. Вот это уже по-мышинному! **LXF**

Спецификации Ryboard	
Процессор Cortex M4 с тактовой частотой 168 МГц и аппаратным модулем для обработки чисел с плавающей запятой	Три 12-битных АЦП (на 16 контактах)
ПЗУ на флэш-памяти объемом 1024 КБ и ОЗУ объемом 192 КБ	Два 12-битных ЦАП (на 2 контактах)
Разъем для подключения карты Micro SD	4 светодиода (красный, зеленый, желтый и голубой)
Трехосный акселерометр (MMA7660)	Стабилизатор напряжения 3,3 В с низким падением напряжения и диапазоном входного напряжения от 3,6 до 10 В
Часы реального времени с дополнительной поддержкой от батареи	Загрузчик прошивки в ПЗУ для удобного обновления прошивки
30 входов/выходов общего назначения (GPIO)	

Уменьшенный Python

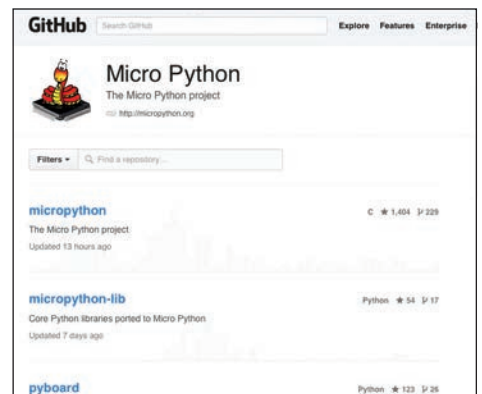
Хотя `Micro Python` — абсолютно новая и отдельная реализация языка `Python`, некоторые все еще борются с тем фактом, что — с точки зрения пользователя — по сути это всё равно `Python` (точнее, `Python 3.3`). Вы точно таким же образом, как в `Python`, вводите код, и результат выводится точно так же.

Естественно, популярность `Python` объясняется его модульностью и тем, что за годы своего существования он оброс библиотеками, которые помогают ему найти конкретное применение в самых разных предметных областях. Хотя большинство библиотек совместимы с `Micro Python` «прямо из коробки», а другие работают с небольшими изменениями, `Micro Python` не был бы по-настоящему «микро», если бы не поставлялся со «швейцарским армейским дробовиком» — стандартной библиотекой.

Так он и делает, и в стандартной поставке есть восемь особенно полезных библиотек: `math`, `cmath` (для комплексных чисел), `gc` для важнейших задач по сбору мусора, `os`, `select`, `struct`, `sys` и `time`.

Разработчики также «уменьшили» несколько компонентов стандартной библиотеки для использования в приложениях `Micro Python`: `urllib`, `urllib2` и `urllib3`. Буква “u” в начале названий библиотек означает, что они были адаптированы для `Micro Python`.

Дополнительные библиотеки можно найти в репозитории `micropython-lib` на <http://github.com/micropython>. Значительное число этих библиотек — «пустышки», а работа над другими еще продолжается, но если вы хотите попробовать их или помочь с разработкой, можем гарантировать, что это будет очень кстати.



➤ Дополнительную информацию и репозиторий библиотек `Python`, портированных для `Micro Python`, см. на [Github](#).

ПЛИС. Много памяти не бывает

Михаил Остапкевич выбирает Papilio Pro, а Евгений Балдин пытается понять, чем ему Papilio One не угодил.



Наш эксперт

Михаил Остапкевич
Романтик, очарованный компьютерами и создаваемыми в них идеальными мирами.



Наш эксперт

Евгений Балдин
Физик, который действительно знает, что такое нехватка вычислительных ресурсов.

Почти все популярные статьи, которые описывают приёмы работы с программируемыми логическими интегральными схемами, начинаются и, увы, заканчиваются рассказом о том, как помигать светодиодом. Признаёмся, что мы тоже отчасти пошли на поводу этой традиции: предыдущая, третья в серии статья (LXF193) была про управление ёлочной гирляндой. Но мы определённо не собираемся на этом заканчивать.

Одной из наших главных целей является обсчёт и визуализация данных больших объёмов. Почему мы этого хотим? Ведь очевидно, что обработку данных с помощью ПЛИС реализовать гораздо сложнее, чем написать классическую последовательную или даже параллельную, с использованием OpenMP или MPI, вычислительную программу. Ответ такой: мы надеемся, что при правильном подходе в итоге удастся получить более производительное решение. Ведь в случае ПЛИС программа фактически воплощается в «железе», или, точнее, в кремнии.

Однако на пути к этой серьёзной цели у нас есть несколько препятствий, и первое из них — это микроскопический объём той памяти, которую мы уже научились использовать в нашем процессоре SPU2014. Объём блоковой памяти на кристалле у схемы Spartan 3E на плате Papilio One 500K составляет около 40 Кбайт. При таком объёме данных нет особого смысла в ускорении, так как они фактически «мгновенно» обрабатываются и с помощью последовательной программы на условно медленном центральном процессоре.

Этот урок про то, как подключить внешнюю память достаточного объёма. Мы рассмотрим механизм управления внешней динамической памятью на примере платы серии Papilio Pro. Выбор этой серии для реализации проекта обусловлен тем, что она является одной из наиболее дешёвых и доступных плат, которые имеют в своем составе отдельные схемы памяти. В рамках Papilio Pro предоставляется доступ к 64 мегабитам и 8 МБ памяти. Для нашего первого прототипа этого вполне достаточно.

По возможности постараемся использовать всё, что разработано ранее, добавив лишь минимум. Возьмём за основу наш процессор SPU2014 и добавим в него возможность работы с внешней памятью DRAM.

Интерфейс с динамическим ОЗУ

Динамическое ОЗУ (DRAM), наряду со статическим — это основные виды оперативных запоминающих устройств. Ячейка памяти в динамическом ОЗУ содержит меньше элементов, чем ячейка статической памяти, и соответственно дешевле. Это позволяет достичь существенно большей плотности хранения данных на кристалле. Иными словами, в итоге микросхемы динамического ОЗУ имеют больший объём памяти и меньшую цену за 1 МБ. Но за это приходится платить: динамическая память медленнее статической, не может работать в широком спектре частот, и у неё выше потребление энергии. С точки же зрения пользователя самое

главное отличие заключается в том, что интерфейс доступа к динамическому ОЗУ фактически на порядок сложнее. Для личной оценки этого различия можно сравнить тот компактный модуль по работе с блоковой памятью (которая, конечно, статическая) из прошлого урока (файл `rppu_mem_ram1.vhd`) с тем, что будет описано далее по тексту.

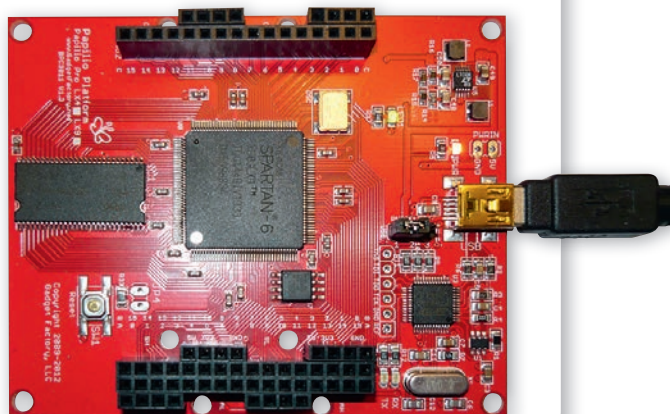
Начнём с описания аппаратной части, а именно — схемы MT48LC4M16 (www.micron.com/~media/documents/products/datasheet/dram/64mb_x4x8x16_sdram.pdf) и её интерфейса с ПЛИС. Схема содержит 64 Мб элементов памяти, из которых сформировано 4М ячеек по 16 бит.

Из 54 выводов микросхемы нас интересуют только те, которые соединяются с ПЛИС:

- » CLK (clock) — входной синхросигнал. Штатно его частота должна быть 133 МГц. При работе в составе Papilio он работает при частотах между 80 и 128 МГц. То есть синхросигнал от кварца на 32 МГц непосредственно использовать нельзя. Его нужно преобразовать, например, в сигнал с частотой 96 МГц.
- » CS# (chip select) — входной сигнал. При подаче нуля схема находится в режиме выполнения поступающих команд.
- » WE#, CAS#, RAS# — входные сигналы, задающие команду для исполнения.

Про Papilio Pro

Плата Papilio Pro (<http://papilio.cc/index.php?n=Papilio.PapilioPro>) является дальнейшим развитием Papilio One. Основные отличия от Papilio One заключаются в более мощном ПЛИС (Xilinx Spartan 6 XC6SLX9), в дополнительном внешнем динамическом ОЗУ на 64 Мбита (СБИС Micron MT48LC4M16) и во флэш-памяти на 64 Мбита (СБИС Macronix MX25L6445). Сам ПЛИС XC6SLX9 содержит 9152 логических ячеек и 576 Кбит блоковой памяти.



» DQM[2] — входной сигнал, определяющий, какие байты адресуемого слова надо перезаписать, а какие не трогать, в операции записи. В случае 16-битных ячеек нужна маска на два байта.

» A[12], BA0, BA1 — входной сигнал, через который вводятся части адреса или содержимое регистра режима. Адрес делится на номер банка, номер ряда и номер колонки.

» DQ[16] — сигнал для передачи битов записываемой или читаемой ячейки (шина данных). Для команды записи это входной сигнал, для чтения — выходной.

В качестве контроллера памяти возьмём реализацию, написанную Майком Филдом [Mike Field] — Simple SDRAM Controller (http://hamsterworks.co.nz/mediawiki/index.php/Simple_SDRAM_Controller). Контроллер реализует память с 32-битными словами. При записи можно с помощью маски избирательно модифицировать только отдельные байты слова. Реализация довольно лаконична. Какой-либо оптимизации при последовательном доступе или при чтении из ранее открытого ряда нет.

Рассмотрим этот интерфейс.

```
COMPONENT SDRAM_Controller is
generic (
  sdram_address_width : natural;
  sdram_column_bits : natural;
  sdram_startup_cycles: natural;
  cycles_per_refresh : natural
);
PORT(
-- подаётся синхросигнал с частотой 96 МГц
  clk : IN std_logic;
-- не пользуемся, всегда 0
  reset : IN std_logic;
-- Сигналы управления контроллером.
-- сигнал от контроллера о том, что он готов принять команду
  cmd_ready : OUT std_logic;
-- сигнал контроллеру, что мы выставляем команду
  cmd_enable : IN std_logic;
-- выбор чтения (0) или записи (1)
  cmd_wr : IN std_logic;
-- 21 бит адреса, указывающего ячейку
-- (22 бита без младшего бита)
  cmd_address : IN std_logic_vector(sdram_address_width-2
  downto 0);
-- указываем при записи, какие байты в слове перезаписывать
  cmd_byte_enable : IN std_logic_vector(3 downto 0);
-- слово для записи в память
  cmd_data_in : IN std_logic_vector(31 downto 0);
-- Данные, получаемые из контроллера.
-- считываемые данные
  data_out : OUT std_logic_vector(31 downto 0);
-- данные прочитаны и выставлены на data_out
  data_out_ready : OUT std_logic;
-- Сигналы, идущие к микросхеме динамической памяти.
-- синхросигнал
  SDRAM_CLK : OUT std_logic;
-- включение использования синхросигнала
  SDRAM_CKE : OUT std_logic;
-- перевод в состояние готовности приема команд
  SDRAM_CS : OUT std_logic;
-- выбор ряда
  SDRAM_RAS : OUT std_logic;
-- выбор столбца
  SDRAM_CAS : OUT std_logic;
-- выбор чтения или записи
  SDRAM_WE : OUT std_logic;
  SDRAM_DQM : OUT std_logic_vector(1 downto 0);
-- шина адреса
```

```
  SDRAM_ADDR : OUT std_logic_vector(12 downto 0);
  SDRAM_BA : OUT std_logic_vector(1 downto 0);
-- шина данных
  SDRAM_DATA : INOUT std_logic_vector(15 downto 0)
);
END COMPONENT;
  Для Papilio Pro выбираем следующие параметры:
-- нужно 22 бита для адресации 4М ячеек constant sdram_ad-
address_width : natural := 22;
constant sdram_column_bits : natural := 8;
-- 100 мкс, плюс ещё чуть-чуть
constant sdram_startup_cycles: natural := 10100;
-- частота обновления памяти
constant cycles_per_refresh : natural := (64000*100)/4196-1;
  На clk подаётся синхросигнал повышенной частоты (подроб-
ности — в следующем разделе). Все сигналы, начиная с SDRAM_
CLK и до конца, нужно подключить к соответствующим контактам
микросхемы ПЛИС. Для управления контроллером задаются сиг-
налы cmd_enable, cmd_wr, cmd_address, cmd_byte_enable и cmd_
data_in, а также считываются сигналы, генерируемые контрол-
лером, cmd_ready, data_out и data_out_ready. Как пользоваться
этими сигналами, будет описано ниже.
```

Синхросигнал для контроллера памяти

Перед началом использования контроллера необходимо обеспе-
чить подачу синхросигнала на него с частотой выше 32 МГц. Нам
подойдёт частота в диапазоне от 80 до 128 МГц.

```
PLL_BASE_inst : PLL_BASE generic map (
-- “HIGH”, “LOW” или “OPTIMIZED”
  BANDWIDTH => “OPTIMIZED”,
-- Задаём коэффициент для умножения частоты (1-64)
  CLKFBOUT_MULT => 24,
-- Сдвиг по фазе не нужен (0.0-360.0).
  CLKFBOUT_PHASE => 0.0,
-- Указываем длительность периода входного сигнала.
-- Для 32 МГц это 31,25 нс.
  CLKIN_PERIOD => 31.25,
-- Задаём делитель частоты для каждого выходного сигнала
(1-128)
  CLKOUT0_DIVIDE => 16, CLKOUT1_DIVIDE => 16,
  CLKOUT2_DIVIDE => 1, CLKOUT3_DIVIDE => 1,
  CLKOUT4_DIVIDE => 1, CLKOUT5_DIVIDE => 1,
-- Задаём скважность сигнала (0.01-0.99).
-- Для указания равных долей '0' и '1' за период
-- используем 0.5
  CLKOUT0_DUTY_CYCLE => 0.5, CLKOUT1_DUTY_CYCLE => 0.5,
  CLKOUT2_DUTY_CYCLE => 0.5, CLKOUT3_DUTY_CYCLE => 0.5,
  CLKOUT4_DUTY_CYCLE => 0.5, CLKOUT5_DUTY_CYCLE => 0.5,
-- Задаём сдвиг по фазе для каждого выходного сигнала
(-360.0-360.0).
  CLKOUT0_PHASE => 0.0, CLKOUT1_PHASE => 0.0,
  CLKOUT2_PHASE => 0.0, CLKOUT3_PHASE => 0.0,
  CLKOUT4_PHASE => 0.0, CLKOUT5_PHASE => 0.0,
-- Выбор источника для CLKFBIN (“CLKFBOUT” или “CLKOUT0”)
  CLK_FEEDBACK => “CLKFBOUT”,
-- “SYSTEM_SYNCHRONOUS”, “SOURCE_SYNCHRONOUS”,
“EXTERNAL”
  COMPENSATION => «SYSTEM_SYNCHRONOUS»,
-- Делитель частоты для всех выходных сигналов (1-52)
  DIVCLK_DIVIDE => 1,
-- Джиттер часов в условных интервалах (0.000-0.999).
-- (фазовое дрожание цифрового сигнала)
  REF_JITTER => 0.1,
-- Всегда FALSE
  RESET_ON_LOSS_OF_LOCK => FALSE
```

»


```

) port map (
-- Подключение выходных сигналов CLKFBOUT, CLKOUT0 -
CLKOUT5 и LOCKED
CLKFBOUT => CLKFB,
CLKOUT0 => CLKu, CLKOUT1 => CLK_MEMu,
CLKOUT2 => open, CLKOUT3 => open,
CLKOUT4 => open, CLKOUT5 => open,
LOCKED => open,
-- Подключение входных сигналов
CLKFBIN => CLKFB, CLKIN => clk, RST => '0'
);
-- Задаём буферизацию синхросигналов
BUFG_1 : BUFG port map (0 => clk, 1 => clk_hw);
BUFG_3 : BUFG port map (0 => clk_dram, 1 => clk);

```

Использование контроллера памяти

Опишем действия, которые необходимо выполнить для записи и чтения с использованием контроллера.

Перед тем, как начинать какую-либо операцию, необходимо дождаться готовности контроллера. Сигнал `cmd_ready` должен перейти в состояние '1'. Если нам нужно произвести запись, то мы задаём адрес ячейки `cmd_address` и записываемое слово `cmd_data_in`, после чего указываем, что это операция записи, установив `cmd_wr` в '1'. Также отмечаем единицами номера байтов в слове, которые надо перезаписать, в 4-битном сигнале `cmd_byte_enable`. Если перезаписывается всё слово целиком, то используется маска "1111". Теперь устанавливаем `cmd_enable` в '1', чтобы сообщить контроллеру о требовании на выполнение нашей команды.

Для чтения последовательность несколько иная. Сначала тоже ждём готовности по сигналу `cmd_ready`, далее указываем адрес в `cmd_address`. Сбрасываем `cmd_wr` в '0' для указания операции чтения. Затем устанавливаем `cmd_enable` в '1'. После этого ждём момента, когда `data_out_ready` переходит в '1'. Это означает, что данные прочитаны и поданы на `data_out`. По окончании считываем данные с `data_out`.

Приведём фрагмент из новой версии модуля `rppu_ctrl_unit.vhd`. В нём обрабатываются состояния выполнения инструкций, которые отвечают за взаимодействие с динамическим ОЗУ:

```

-- состояние начала считывания кода инструкции из памяти
when mem_waitready_fetch =>
-- ждём готовности контроллера памяти обрабатывать запрос
if mem_ready = '1' then
-- устанавливаем адрес кода инструкции для считывания
reg_mem_addr <= reg_ip;
-- задаём операцию чтения
reg_mem_r_w <= '0';
-- выставляем запрос к контроллеру памяти
reg_mem_req <= '1';
reg_mem_busy <= '1';
iexec_state <= mem_start_r;
end if;
-- состояние начала считывания данных из памяти
when mem_waitready_r =>
-- ждём готовности контроллера памяти обрабатывать запрос
if mem_ready = '1' then
-- устанавливаем адрес данных для считывания
reg_mem_addr <= reg_acc;
-- задаём операцию чтения
reg_mem_r_w <= '0';
-- выставляем запрос к контроллеру памяти
reg_mem_req <= '1';
reg_mem_busy <= '1';
iexec_state <= mem_start_r;
end if;

```

```

-- состояние начала записи данных в память
when mem_waitready_w =>
-- ждём готовности контроллера памяти обрабатывать запрос
if mem_ready = '1' then
-- устанавливаем адрес данных для записи
reg_mem_addr <= reg_acc;
-- устанавливаем содержимое для записи
reg_mem_data_write <= reg_aux;
-- задаём операцию записи
reg_mem_r_w <= '1';
-- выставляем запрос к контроллеру памяти
reg_mem_req <= '1';
reg_mem_busy <= '1';
iexec_state <= mem_start_w;
end if;
-- начало записи
when mem_start_w =>
-- ждём сигнала, что обработка запроса началась
if mem_ready = '0' then
-- сбрасываем линию запроса
reg_mem_req <= '0';
iexec_state <= mem_acked;
end if;
when mem_start_r =>
-- ждём сигнала, что данные считаны
if mem_data_ready = '1' then
-- сбрасываем линию запроса
reg_mem_req <= '0';
iexec_state <= mem_acked;
if iexec_fetch = '1' then
-- если считывали код, то запишем его в регистр кода
-- инструкции
reg_code <= mem_data_read;
else
-- если считывали данные, то запишем их в регистр
-- результата
reg_res <= mem_data_read;
end if;
end if;
when mem_acked =>
-- ждёмся, что память готова обработать следующий
-- запрос
if mem_ready = '1' then
-- в более производительной версии мы исключим
-- это состояние
iexec_state <= mem_finish;
end if;
when mem_finish =>
reg_mem_busy <= '0';
if iexec_fetch = '1' then
iexec_fetch <= '0';
iexec_state <= start;
else
iexec_state <= finish;
end if;

```

Чтобы стала понятна связь данного исходника с интерфейсом контроллера SDRAM_Controller, ниже приведём фрагмент исходника `rppu_assembly.vhd`, где сигналы из этого исходника и интерфейса связываются. Это простейшая версия для 8-битного адреса и 8-битных ячеек памяти. Она неэффективна и использует малую часть памяти, но проста в понимании. Более сложные, но и более полезные версии реализации интерфейса можно посмотреть на сайте <http://qwertus.com/fpga/p4.html>.

```

cmd_enable <= wire_ram_launch;
cmd_wr <= wire_ram_rw;

```

```

cmd_address <= "000000000000" & wire_ram_addr;
cmd_data_in <= "000000000000000000000000" &
wire_ram_data_write;
cmd_byte_enable <= "1111";
wire_ram_data_read <= data_out(7 downto 0);
wire_ram_ready <= cmd_ready;
wire_ram_data_ready <= data_out_ready;
Inst_SDRAM_Controller: SDRAM_Controller GENERIC MAP (
  sdr_ams_address_width => sdr_ams_address_width,
  sdr_ams_column_bits => sdr_ams_column_bits,
  sdr_ams_startup_cycles => sdr_ams_startup_cycles,
  cycles_per_refresh => cycles_per_refresh
) PORT MAP(
  clk => clk_dram,
  reset => '0',
  cmd_address => cmd_address,
  cmd_wr => cmd_wr,
  cmd_enable => cmd_enable,
  cmd_ready => cmd_ready,
  cmd_byte_enable => cmd_byte_enable,
  cmd_data_in => cmd_data_in,
  data_out => data_out,
  data_out_ready => data_out_ready,
  SDRAM_CLK => SDRAM_CLK,
  SDRAM_CKE => SDRAM_CKE,
  SDRAM_CS => SDRAM_CS,
  SDRAM_RAS => SDRAM_nRAS,
  SDRAM_CAS => SDRAM_nCAS,
  SDRAM_WE => SDRAM_nWE,
  SDRAM_DQM => SDRAM_DQM,
  SDRAM_BA => SDRAM_BA,
  SDRAM_ADDR => SDRAM_ADDR,
  SDRAM_DATA => SDRAM_DQ
);

```

Доработаем наш файл ограничений, включив в него объявления для всех сигналов, требующихся для работы с памятью (в частности, 12 бит адреса, 16 бит данных). Также изменим определения для сигналов TX, RX и выводов, которые мы в прошлый раз использовали для подключения светодиодов — они на плате Papilio Pro подключены иначе, чем на Papilio One 500K. В результате у нас получается следующее:

```

## Запрещаем автоматическое размещение выводов,
подсоединенных к питанию и земле.
CONFIG PROHIBIT=P144;
CONFIG PROHIBIT=P69;
CONFIG PROHIBIT=P60;
NET "clk" LOC="P94" | IOSTANDARD=LVTTL | PERIOD=31.25ns;
# CLK
NET p17 LOC="P134" | IOSTANDARD=LVTTL; # C15
NET p16 LOC="P133" | IOSTANDARD=LVTTL; # C14
NET p15 LOC="P132" | IOSTANDARD=LVTTL; # C13
NET "rx" LOC="P101" | IOSTANDARD=LVTTL | DRIVE=8 |
SLEW=FAST;
NET "tx" LOC="P105" | IOSTANDARD=LVTTL | DRIVE=8 |
SLEW=FAST | PULLUP;
NET SDRAM_ADDR(0) LOC="P140" | IOSTANDARD=LVTTL;
# SDRAM_ADDR0
NET SDRAM_ADDR(1) LOC="P139" | IOSTANDARD=LVTTL;
# SDRAM_ADDR1
NET SDRAM_ADDR(2) LOC="P138" | IOSTANDARD=LVTTL;
# SDRAM_ADDR2
NET SDRAM_ADDR(3) LOC="P137" | IOSTANDARD=LVTTL;
# SDRAM_ADDR3
NET SDRAM_ADDR(4) LOC="P146" | IOSTANDARD=LVTTL;
# SDRAM_ADDR4

```

```

NET SDRAM_ADDR(5) LOC="P45" | IOSTANDARD=LVTTL;
# SDRAM_ADDR5
NET SDRAM_ADDR(6) LOC="P44" | IOSTANDARD=LVTTL;
# SDRAM_ADDR6
NET SDRAM_ADDR(7) LOC="P43" | IOSTANDARD=LVTTL;
# SDRAM_ADDR7
NET SDRAM_ADDR(8) LOC="P41" | IOSTANDARD=LVTTL;
# SDRAM_ADDR8
NET SDRAM_ADDR(9) LOC="P40" | IOSTANDARD=LVTTL;
# SDRAM_ADDR9
NET SDRAM_ADDR(10) LOC="P141" | IOSTANDARD=LVTTL;
# SDRAM_ADDR10
NET SDRAM_ADDR(11) LOC="P35" | IOSTANDARD=LVTTL;
# SDRAM_ADDR11
NET SDRAM_ADDR(12) LOC="P34" | IOSTANDARD=LVTTL;
# SDRAM_ADDR12
NET SDRAM_DQ(0) LOC="P9" | IOSTANDARD=LVTTL; # SDRAM_DQ0
NET SDRAM_DQ(1) LOC="P10" | IOSTANDARD=LVTTL; # SDRAM_DQ1
NET SDRAM_DQ(2) LOC="P11" | IOSTANDARD=LVTTL; # SDRAM_DQ2
NET SDRAM_DQ(3) LOC="P12" | IOSTANDARD=LVTTL; # SDRAM_DQ3
NET SDRAM_DQ(4) LOC="P14" | IOSTANDARD=LVTTL; # SDRAM_DQ4
NET SDRAM_DQ(5) LOC="P15" | IOSTANDARD=LVTTL; # SDRAM_DQ5
NET SDRAM_DQ(6) LOC="P16" | IOSTANDARD=LVTTL; # SDRAM_DQ6
NET SDRAM_DQ(7) LOC="P8" | IOSTANDARD=LVTTL; # SDRAM_DQ7
NET SDRAM_DQ(8) LOC="P21" | IOSTANDARD=LVTTL; # SDRAM_DQ8
NET SDRAM_DQ(9) LOC="P22" | IOSTANDARD=LVTTL; # SDRAM_DQ9
NET SDRAM_DQ(10) LOC="P23" | IOSTANDARD=LVTTL;
# SDRAM_DQ10
NET SDRAM_DQ(11) LOC="P24" | IOSTANDARD=LVTTL;
# SDRAM_DQ11
NET SDRAM_DQ(12) LOC="P26" | IOSTANDARD=LVTTL;
# SDRAM_DQ12
NET SDRAM_DQ(13) LOC="P27" | IOSTANDARD=LVTTL;
# SDRAM_DQ13
NET SDRAM_DQ(14) LOC="P29" | IOSTANDARD=LVTTL;
# SDRAM_DQ14
NET SDRAM_DQ(15) LOC="P30" | IOSTANDARD=LVTTL;
# SDRAM_DQ15
NET SDRAM_DQM(0) LOC="P7" | IOSTANDARD=LVTTL;
# SDRAM_DQML
NET SDRAM_DQM(1) LOC="P17" | IOSTANDARD=LVTTL;
# SDRAM_DQMH
NET SDRAM_BA(0) LOC="P143" | IOSTANDARD=LVTTL; # SDRAM_BA0
NET SDRAM_BA(1) LOC="P142" | IOSTANDARD=LVTTL; # SDRAM_BA1
NET SDRAM_nWE LOC="P6" | IOSTANDARD=LVTTL; # SDRAM_nWE
NET SDRAM_nCAS LOC="P5" | IOSTANDARD=LVTTL; # SDRAM_nCAS
NET SDRAM_nRAS LOC="P2" | IOSTANDARD=LVTTL; # SDRAM_nRAS
NET SDRAM_CS LOC="P1" | IOSTANDARD=LVTTL; # SDRAM_CS
NET SDRAM_CLK LOC="P32" | IOSTANDARD=LVTTL; # SDRAM_CLK
NET SDRAM_CKE LOC="P33" | IOSTANDARD=LVTTL; # SDRAM_CKE

```

Заключение

На этом уроке мы доработали наш процессор. Теперь он может использовать для хранения своих программ и данных внешнее динамическое ОЗУ. Причём для описанного случая с платой Papilio Pro объём памяти вырастает до 8 МБ. Для более мощных плат он может быть существенно выше. Например, для платы Xilinx KC705 это 1 Гб. Полный набор исходных текстов процессора SPU2015v1 и архива с проектом для ISE WebPack доступен на <http://qwertus.com/fpga/p4.html>.

На диске к журналу прилагается архив с исходниками на VHDL для 16-битной версии проекта (один из исходников написан М. Филдом).

На следующем уроке SPU2015v1 будет дополнен вычислительным сопроцессором. **LXF**

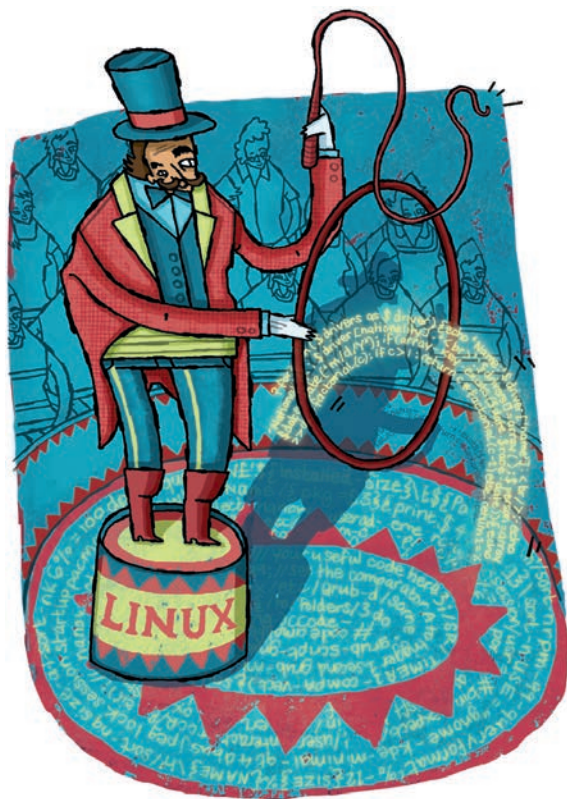
Matter.js: Пишем HTML5-игры

Лайам Браммит объясняет, как с помощью его физического движка, Matter.js, создать простую, но захватывающую игру на основе технологии HTML5.



Наш эксперт

Лайам Браммит — разработчик CSS3, HTML5 и JavaScript, и если ему снятся кошмары, то их тема — «JavaScript выбрасывает сообщение „Требуется IE6“».



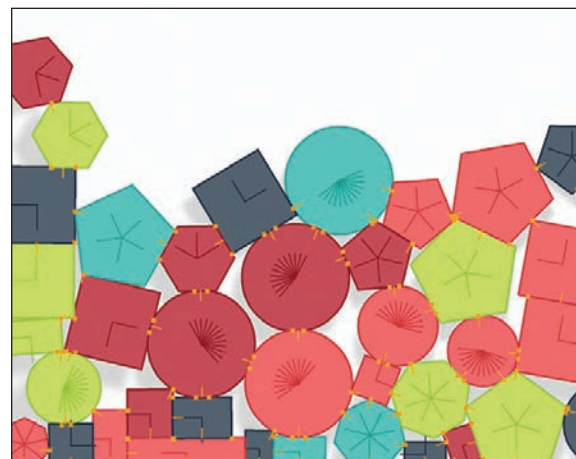
Matter.js — гибкий солидный 2D-движок с упором на высокую производительность, стабильность, простоту и кроссплатформенную совместимость. Он не является полностью игровым движком — скорее его компонентом, но включает большинство вещей, необходимых для старта, включая игровой цикл, контроллер мыши, систему столкновений и рендерер. На данный момент проект находится в стадии альфа и имеет все обычные недостатки болезней роста — например, его не стоит применять в рабочей среде; но тем не менее хорошо подходит для независимых двумерных игр, которым не нужен сверхмощный движок.

«Да у меня уже есть движок WhizzBang200», скажете вы. Не проблема. Так как Matter.js включает систему обратных вызовов событий и настраиваемые контроллеры, у вас должно получиться интегрировать физический движок в любой игровой движок JavaScript или библиотеку рендеринга, которыми вы пользуетесь. Движок включает пример рендера «на двумерном холсте» с базовой поддержкой рендеринга как примитивов (векторного), так и спрайтов (текстурного) — физических объектов игрового мира. Решив двинуться дальше, вы легко сможете использовать эти рендеры как основу для своего собственного. А если холсты слишком прозрачны для вашего вкуса

и вы предпочитаете новейшие возможности WebGL, то вам тоже повезло: замените в этом примере рендерер на WebGL (который использует прекрасную библиотеку Pixi.js), и все готово. Этот графический рендерер сцен можно расширить, чтобы воспользоваться и чем-то вроде three.js.

Несмотря на крохотный размер (менее 200 КБ или 53 КБ в урезанном варианте), Matter.js удается вмещать массу сложных технологий игровой физики, и к тому же здесь всё это гораздо удобнее, чем во многих более крупных проектах. В частности, библиотека включает:

- » Интегрирование Верле [Verlet] с коррекцией положения по времени.
- » Обнаружение по широкой фазе с адаптивной сеткой.
- » Обнаружение по средней фазе в параллелепипеде со сторонами, параллельными осям координат.
- » Обнаружение по узкой фазе в SAT.
- » Итеративный разрешитель последовательности импульсов и положения.
- » Разрешение столкновений с ограничениями, аналогично методу Эрина Катто [Erin Catto].
- » Кэширование и разогрев импульса временной когерентности.
- » Пары столкновений, контакты и импульсы, поддерживаемые менеджером пар.
- » Приближенная модель трения Кулона [Coulomb] с ограничениями трения.
- » Разрешение ограничений методом Гаусса–Зиделя [Gauss–Siedel].
- » Полупеременный временной шаг, синхронизированный с рендерингом.
- » Базовая стратегия засыпания.
- » Рендерер холста HTML5 / WebGL.



» Matter.js — хорошо масштабируемый движок. Это означает, что он может работать с множеством имитируемых объектов.

Скорая помощь

Более подробную информацию о Matter.js можно получить на странице проекта — <http://brm.io/matter-js>; там же вы найдете несколько примеров и исчерпывающую документацию.

Пишем простую игру

Пора показать, как же воспользоваться Matter.js. Мы создадим простую сцену: пирамиду из блоков, метательную пращу и камень, которым мы выстрелим и разрушим пирамиду. Если вы застрянете или что-то покажется непонятным, на странице проекта (<http://brm.io/matter-js>) всегда можно заглянуть в примеры движков и документацию по API. Начнем по порядку: скопируйте движок с диска или загрузите его с домашней страницы проекта и создайте чистую HTML-страницу со ссылкой на Matter.js в тэге `<script>`.

```
<html>
<script src=matter-0.8.0.js type="text/javascript">
</script>
```

При желании можно воспользоваться и минимальной версией, **matter-0.8.0.js** — для наших целей подойдет обе.

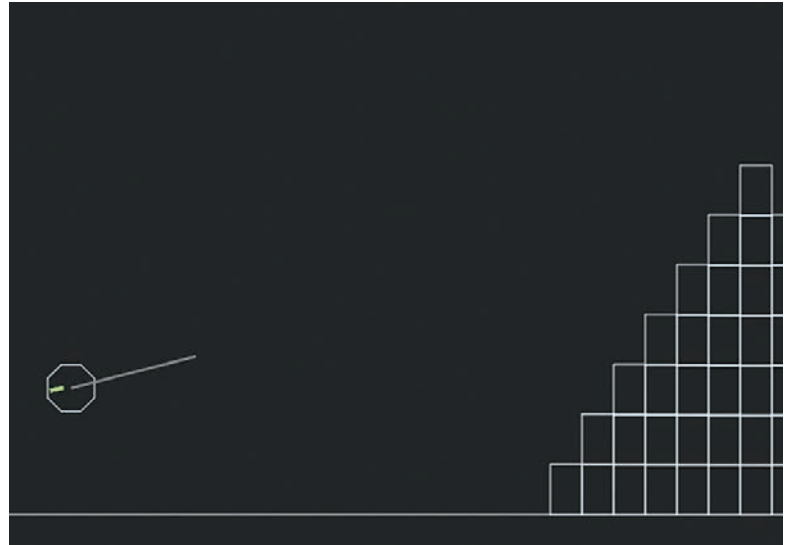
Откройте еще один блок `script`, чтобы затем добавить в нее код, указанный ниже. Почти весь оставшийся код разместится в функции `init`, которую мы объявим так:

```
<script>
<script>
// Matter.js - http://brm.io/matter-js/
function init() {
```

Мы опустили один уровень отступов в оставшемся коде, чтобы он легче читался. Не забывайте, что всё это — внутри функции `init()`.

При включении Matter.js объект **Matter** становится доступным глобально. В нем и находятся все функции Matter. Чтобы всё было лаконично, стоит создать алиасы для необходимых модулей в верхней части скрипта таким образом:

```
var Engine = Matter.Engine,
World = Matter.World,
Bodies = Matter.Bodies,
Constraint = Matter.Constraint,
```



```
Composites = Matter.Composites,
MouseConstraint = Matter.MouseConstraint,
Events = Matter.Events;
```

Сначала создадим движок, передав ему элемент, в который будет вставлен холст:

```
var engine = Engine.create(document.body);
```

Мы хотим управлять сценой с помощью мыши. В движке есть удобное ограничение `MouseConstraint`, поэтому добавим его:

```
var mouse = MouseConstraint.create(engine, { constraint:
{stiffness: 1 }
});
```

Здесь мы передали движок, который требуется для работы **MouseConstraint**. Я заодно передал еще и объект с параметрами, »

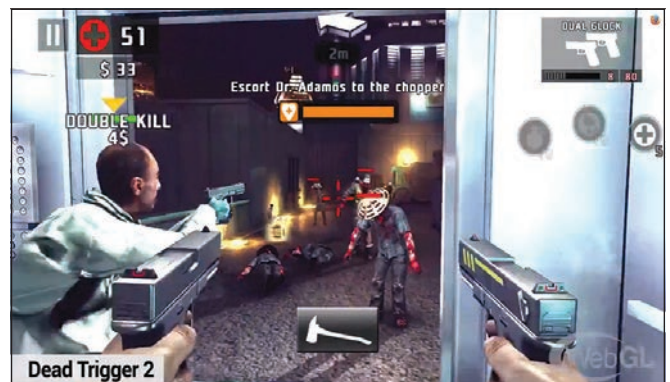
» Для простой демонстрации создадим в этом руководстве игру с пращой.

Зачем писать на HTML5 игры с физикой?

Популярность приложений с игровой физикой на рынках мобильных приложений говорит о том, что физика часто является важным компонентом современных игр, а во многих играх — ключевым. Почему? Похоже, есть некое наслаждение в создании и разрушении чего-нибудь! Несколько лет назад идея создания настоящих игр на основе web-технологий показалась бы безумной. К счастью,

web-технология становится жизнеспособным инструментом размещения контента на нескольких платформах без (особых) усилий. В последние годы произошло существенное улучшение технологий HTML5, от производительности основных движков JavaScript (V8, SpiderMonkey и т. д.) до API для холста и WebGL. И дальше все только улучшается. Недавно Mozilla поработала с Unity (netm.ag/unitywe-

[bgl-255](http://netm.ag/unrealwebgl-255)) и Unreal Engine (netm.ag/unrealwebgl-255), чтобы они с присущей им скоростью работали в браузере с помощью `asm.js`, высокооптимизированной версии JavaScript. Это впечатляет. Matter.js, может, и выбивается из данного списка — это не полноценный игровой движок вроде Unity и Unreal — и тем не менее хорошо подойдет для авторских 2D-игр: мощный движок им ни к чему.



» В пятой версии популярного игрового движка Unity добавлен экспорт WebGL. А у движка Unreal есть демо-игры в сети, например, *Tappy Chicken*.

» Не хотите пропустить номер? Подпишитесь на www.linuxformat.ru/subscribe!

в котором изменен параметр **stiffness** ограничения, чтобы оно было жестче по сравнению со значением по умолчанию. Добавим к сцене несколько объектов. Прежде всего, нам нужен объект «земля», иначе другие объекты выпадут из мира!

```
var ground = Bodies.rectangle(395, 600, 815, 50, { isStatic: true });
```

Здесь модуль **Bodies** — фабрика, способная создавать геометрические объекты различных типов: квадраты, прямоугольники, окружности, треугольники и другие многоугольники. Я передал ей необходимые нам положение и размерности, а также снова передал объект с параметрами. На сей раз мы помечаем землю как статический объект (**isStatic**), так как обычно она неподвижна.

Создаем физические объекты

Теперь создадим камень. Будем оригинальными — пусть камень будет шестиугольным с радиусом 20 пикселей:

```
var rock = Bodies.polygon(170, 450, 8, 20);
```

Чтобы из пращи можно было выстрелить, нам понадобится эластичная веревка-жгут. Для этого воспользуемся **Constraint**:

```
var anchor = { x: 170, y: 450 },
```

```
elastic = Constraint.create({ pointA: anchor, bodyB: rock, stiffness: 0.1 });
```

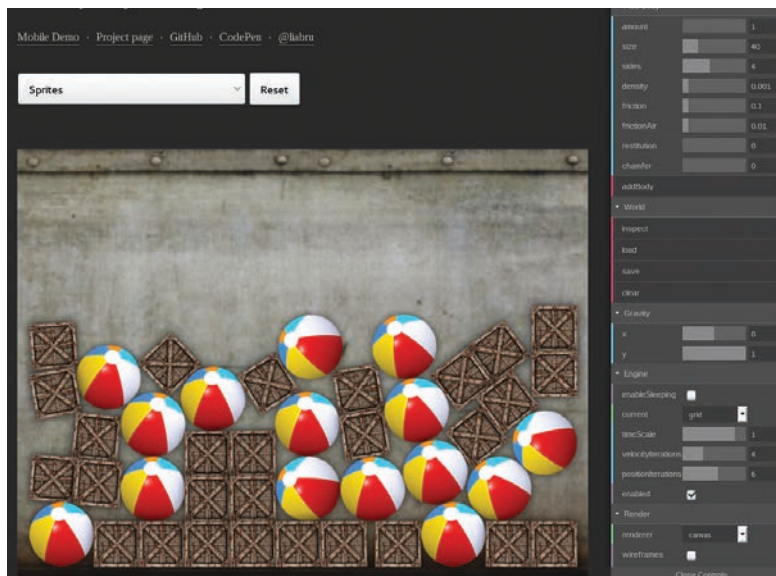
Как видите, один конец **Constraint** будет закреплен в **pointA**, якорю с фиксированным положением в сцене. Другой конец закреплен в центре только что созданного камня. Я сделал **stiffness** довольно низким, чтобы ограничение было более эластичным и мы бы получили немного ударной мощи для выстрела из своего метательного орудия.

Затем нам нужна цель для поражения. Движок поддерживает композитные объекты (**Composite**), которые представляют собой сочетания **Body**, **Constraint** и даже других дочерних объектов **Composite**, которыми можно манипулировать вместе как группой. Нам нужна пирамида из блоков, для создания которой обычно требуется довольно много кода. Однако, поскольку это расхожее требование, Matter.js включает фабрику **Composites**, с помощью которой легко создавать стопки и пирамиды:

```
var pyramid = Composites.pyramid(450, 300, 13, 10, 0, 0,
function(x, y, column, row) {
return Bodies.rectangle(x, y, 25, 40);
});
```

Не буду перебирать все аргументы — их легко найти в документации по API на странице проекта. (Вкратце, фабрика пирамид использует функцию обратного вызова, в которой вы создаете объекты, и затем корректно складывает их в стопку.) Теперь у нас есть все необходимые объекты и ограничения — давайте же добавим их в мир:

```
World.add(engine.world, [mouse, ground, pyramid, rock, elastic]);
```



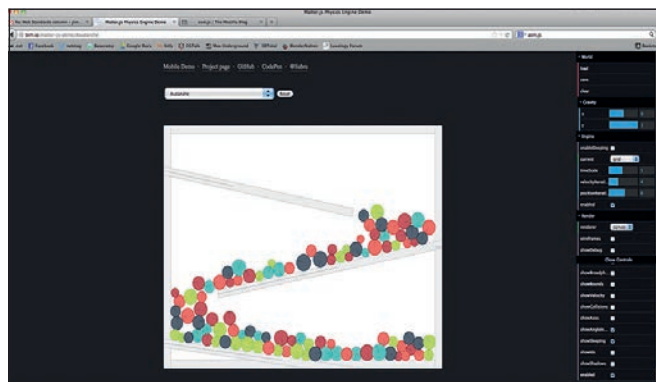
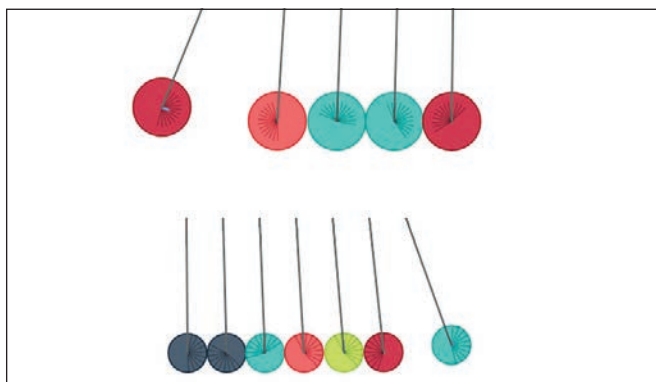
➤ Демо Sprites показывает, что все не обязательно должно сводиться к сплошным цветам и ячейкам.

Демонстрационные сцены

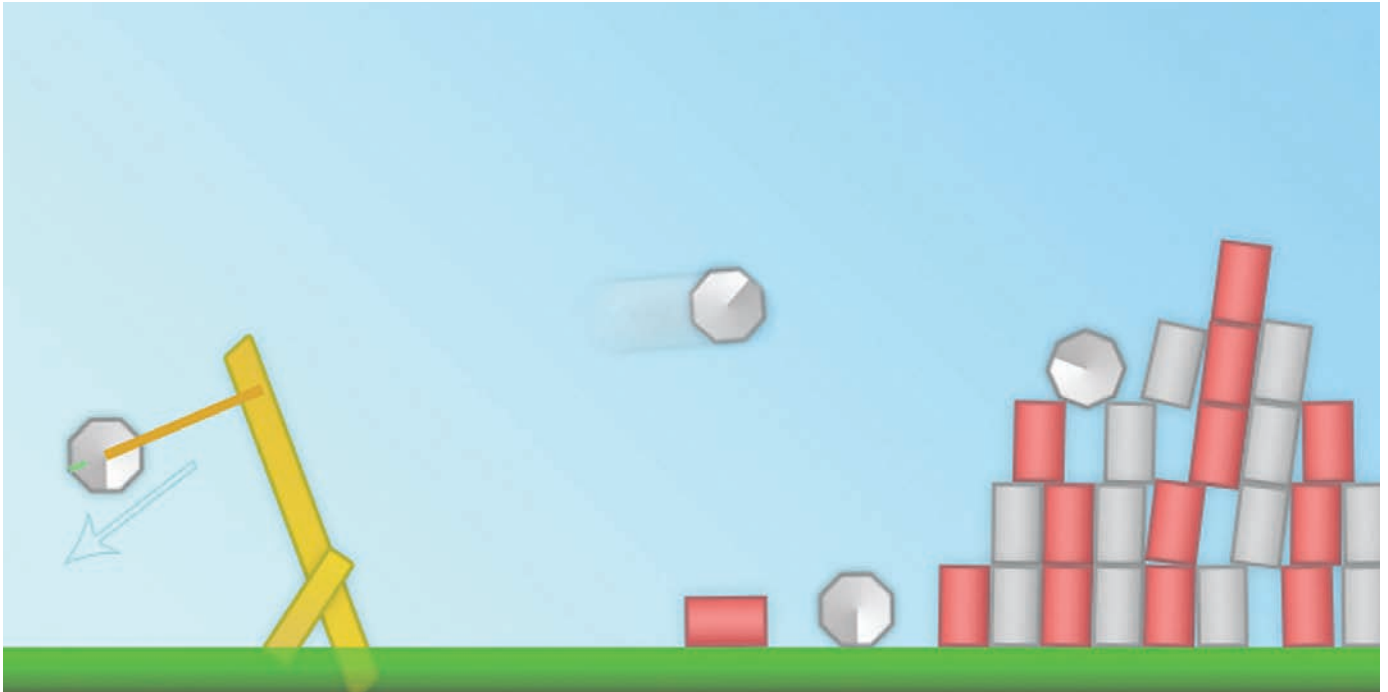
На данный момент на сайте проекта Matter.js (<http://brm.io/matter-js>) можно найти двадцать различных примеров и демо, в том числе классическую колыбель Ньютона (на рис. ниже), которая — наряду с тем, что слегка гипнотизирует — демонстрирует свойство сохранения импульса. Также есть демо

для мобильных устройств, которое включает код сдвига гравитации на основе данных датчиков наклона устройства. Хотя на данный момент это хорошо работает только на более дорогих устройствах, потенциал для мобильных устройств заметен даже на этой ранней стадии. В демо на сайте проекта

используется простая черно-белая контурная графика, но вы можете увидеть и более красочную версию демо с гравитацией на сайте CodePen (<http://codepen.io/ljabru>) или зайти в полноценное онлайн-демо (<http://brm.io/matter-js-demo>) и поиграть с настройками рендеринга самостоятельно.



➤ Хотите исследовать все доступные параметры? Сделайте это в полноценном онлайн-демо Matter.js.



Задание игровой логики

Последний этап — добавить немного игровой логики. Когда игрок сделал один выстрел, нам нужно отсоединить камень от жгута и загрузить в пращу новый камень. Для этого мы привяжем к движку событие таймера (тик) вызывается при каждом обновлении движка. Обычно это подразумевает 60 тиков в секунду (при частоте в 60 кадров в секунду):

```
Events.on(engine, 'tick', function(event) {
  if (engine.input.mouse.button === -1 && rock.position.x > 190) {
    rock = Bodies.polygon(170, 450, 7, 20)
    World.add(engine.world, rock); elastic.bodyB = rock;
  }
});
```

Приведенный выше код делает несколько вещей. Во-первых, он проверяет, что не нажата ни одна из кнопок мыши, а если это так, проверяет, переместился ли камень на жгуте по горизонтали относительно своего исходного положения покоя. Тогда камень нужно отпустить. В дальнейшем коде создается новый камень

и связывается с ограничением жгута. Это означает, что предыдущий камень больше не привязан к жгуту и отпущен и, надеюсь, летит в нашу пирамиду из блоков. Более того, теперь в праще новый камень, который готов к броску. Теперь все готово; последнее, что нужно сделать — запустить симуляцию и закрыть функцию `init()`:

```
Engine.run(engine);
}
```

Очень важно позаботиться, чтобы после объявления функции `init` вы привязали ее к событию загрузки окна — в противном случае вы ничего не увидите. Для этого добавьте следующий код сразу после функции `init()`:

```
window.addEventListener('load', init);
```

Не забудьте добавить закрывающие теги `</script>` и `</html>`. Когда вы откроете страницу в браузере, на ней должна появиться игра, готовая к запуску. Если она не работает как должно, проверьте консоль, затем откройте ссылку с кодом руководства и сравните его со своим. **LXF**

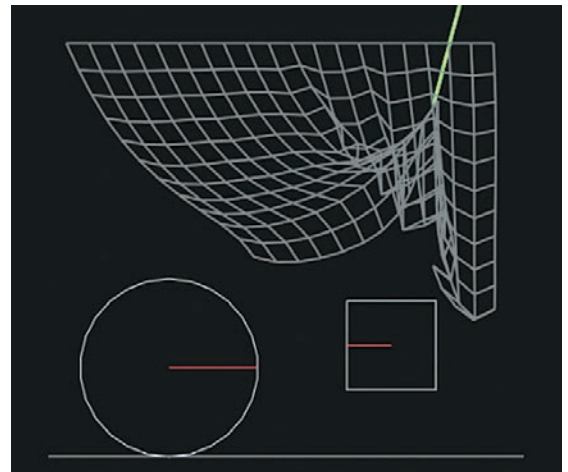
➤ Хотя это не совсем *Angry Birds*, превращать пирамидки в строительный мусор все равно довольно приятно. В *Matter.js* используются примерно те же технологии симуляции объектов, что и в *Box2D*, движке, используемом в игре о сердитых птичках и во многих других.

О ТКАНИ

Пример с тканью — один из многих, которые можно найти на демонстрационной странице. Отдельные точки ткани можно перетаскивать, и она будет падать вниз под влиянием гравитации и окружности и квадрата, которые встретятся ей на пути. Наша ткань — из 19×11 эластичных ячеек (эта клеточная структура образуется после прохождения 20 строк и 12 столбцов через объект класса `softBody`), которые вместе эмулируют ее мягкую поверхность. Объект `Composites` в *Matter.js* абстрагирует всю сложность симуляции, поэтому если не считать разнородного кода установки параметров (частично его можно найти ранее в этом руководстве, а оставшуюся часть — на сайте), то рабочая часть кода довольно проста:

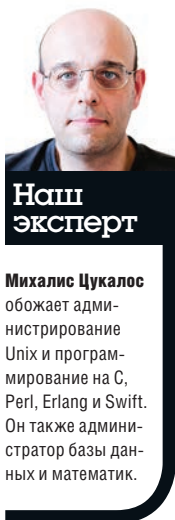
```
particleOptions = { friction: 0.00001, groupId:
  groupId, render: { visible: false }},
cloth = Composites.softBody(200, 200, 20,
  12, 5, 5, false, 8, particleOptions);
for (var i = 0; i < 20; i++) { cloth.bodies[i].
  isStatic = true; }
World.add(world, [
  cloth,
  Bodies.circle(300, 500, 80, { isStatic: true }),
  Bodies.rectangle(500, 480, 80, 80, { isStatic:
  true })
]);
```

Интерактивность мыши прилежно обрабатывается модулем `MouseConstraint`, который любезно учитывает и нажатия клавиш мыши. *Matter.js* также поддерживает некоторые мобильные функции, например, ввод от датчика наклона.



Erlang: УЧИМСЯ НА ПРАКТИКЕ

Михалис Цукалос помогает начать программировать на Erlang. Сразу же забудьте все, что знаете: нас ждет функциональное программирование!..



Erlang — открытый язык функционального программирования, разработанный в Лаборатории вычислительных наук Ericsson для обслуживания коммутации телефонного оборудования. Хотя кажется, что это язык программирования специального назначения, это не так. Однако функциональное программирование — совершенно другая парадигма программирования. Вы должны забыть большинство из того, что уже знаете

о процедурном или объектно-ориентированном программировании, и начать думать по-новому. И Erlang такой не один — это необходимо и для других языков функционального программирования: Elixir, SML, Scala и Idris.

Проектирование и программирование на языке Erlang следует ряду базовых принципов. Первый принцип — системы склонны «падать», и избежать этого нельзя... ну так пусть «падают»! А вот чего избежать можно — это необнаружения ошибки, и вы обязаны ее обнаружить. Но одного обнаружения недостаточно: надо суметь определить причину сбоя. Другой принцип — вычисления должны быть независимы, чтобы использовать несколько процессоров и ядер, не утруждаясь разделяемыми переменными и памятью. Тут имеется очень желательный побочный эффект: код лучше масштабируется! (На сайте www.erlang.org Erlang именно и описан как «обширно масштабируемый».) Процессы в Erlang общаются друг с другом посредством сообщений; и это хорошо, так как в Erlang сообщения передаются очень быстро.

Erlang также позволяет программировать web-сервисы с помощью библиотеки MochiWeb.

Числа и переменные в Erlang

Переменные в Erlang начинаются с Заглавной Буквы, а «слова», начинающиеся со строчной буквы, представляют собой Атомы. Атомы — это константы, имя которых служит их значением.

После объявления переменной в оболочке Erlang изменить ее значение нельзя, что иллюстрирует следующий пример:

```
1> MyVar = 12.
12
2> MyVar = MyVar + 5.
** exception error: no match of right hand side value 17
[ошибка: несоответствие величине 17 в правой части]
3> MyVar = 5.
** exception error: no match of right hand side value 5
[ошибка: несоответствие величине 5 в правой части]
```

Что такое OTP

OTP — аббревиатура от “Open Telecom Platform [Открытая телекоммуникационная платформа]”, и этим именем называется также набор открытых библиотек и утилит Erlang. Несмотря на своё отчасти ограничительное название, OTP — отнюдь не только телефоны!

OTP была разработана для создания больших проектов большими командами. OTP означает «взять все общие компоненты, поместить их в библиотеки, убедившись, что они работают правильно

и надежно, и затем использовать этот код повторно как можно чаще».

Что же остается программисту? Ему остается заниматься тем, что изменяется от приложения к приложению, а OTP позаботится обо всем стандартном. И если вы собираетесь писать на Erlang настоящие программы, вам обязательно придется изучить фреймворк OTP.

OTP позволяет надзирать за выполнением существующего кода на Erlang. Чтобы наблюдать

за существующим модулем, придется написать дополнительный модуль на Erlang, но в существующем модуле при этом нельзя ничего менять! Если наблюдающий модуль увидит, что серверный процесс дал сбой, он автоматически перезапустит серверный процесс. OTP — самая важная и незаменимая часть Erlang.

Более подробный рассказ об OTP лежит за рамками нашего урока. Мы расскажем об OTP в отдельной статье — она появится уже скоро.

В Erlang используется модель однократного присваивания, позволяющая присваивать значение переменной только один раз в заданной сессии или контексте. Поэтому Erlang интерпретирует “MyVar = MyVar + 5” как “12 = 12 + 5”, что явно неверно с математической точки зрения.

Erlang не похож на другие, производные от C языки программирования. В Erlang программирование больше сосредоточено на потоке данных, а значит, вам придется научиться проектировать свои программы с использованием рекурсии и передачи сообщений.

Встроенный редактор строк в оболочке Erlang (известный как *erl*) представляет собой подмножество *Emacs*. В оболочке Erlang можно протестировать большую часть кода перед созданием модуля. Прежде чем читать дальше, запомните, что для выхода из оболочки Erlang достаточно набрать **q()**, и нажать Enter. Точка в конце обязательна.

Код на Erlang для программы “Hello World”, заключенной в модуле Erlang, будет следующим:

```
-module(helloWorld).
-export([hello_world/0]).
hello_world() -> io:fwrite("Hello World!\n").
```

Выражение **export([hello_world/0])** показывает, что функция **hello_world()** не принимает аргументов. Первая команда объявляет имя модуля.

Теперь можно запустить программу в оболочке Erlang следующим образом (файл **helloWorld.erl** должен находиться в текущем рабочем каталоге):

```
$ erl
Erlang R15B01 (erts-5.9.1) [source] [64-bit] [async-threads:0]
[kernel-poll:false]
Eshell V5.9.1 (abort with ^G)
1> c(helloWorld) %% No dot at the end of the command
[Нет точки в конце команды]
1> c(helloWorld). %% Interpreted as “c(helloWorld) c(helloWorld).”
[Интерпретировалось как]
* 2: syntax error before: c
1> c(helloWorld). %% Now it is right [Теперь правильно]
{ok,helloWorld} %% Compilation was OK.[Компиляция успешна]
2> helloWorld:hello_world(). %% Run the desired function
[Запуск требуемой функции]
Hello world!
ok
```

Каждая команда, выполняемая в оболочке Erlang, должна заканчиваться точкой. Команда **c(helloWorld)** компилирует исходный код, найденный в файле **helloWorld.erl**. Команда **helloWorld:hello_world()** вызывает функцию **hello_world()**, которую можно найти в модуле **helloWorld**. При выполнении команды **io:fwrite(«Hello World!\n»)** в оболочке, функция **fwrite** выводит нужное сообщение, а оболочка Erlang выводит возвращаемое значение (“ok”). Также можно скомпилировать файл **helloWorld.erl** и запустить нужную функцию из оболочки Linux:

```
$ erl -compile helloWorld.erl
$ ls -l helloWorld.*
-rw-r--r-- 1 mtsouk staff 588 Oct 13 19:46 helloWorld.beam
-rw-r--r--@ 1 mtsouk staff 93 Oct 13 19:42 helloWorld.erl
$ erl -noshell -s helloWorld hello_world -s init stop
Hello world!
```

Функции в Erlang (как и в любом другом языке функционального программирования) — объекты первого класса, и их можно использовать как аргументы других функций. Функции также могут возвращать другие функции! Erlang поддерживает *Funs*, функциональные объекты. Они позволяют создавать анонимные

объекты и передавать функцию как аргумент другой функции. Другие типы структур данных, поддерживаемые в Erlang — *Tuple* [кортеж], *Map* [карта], *List* [список] и *Record* [запись].

В Erlang нет цикла **for**, да он здесь и не нужен — его можно заменить шаблонами и функциями более высокого порядка. Тем не менее, цикл **for** можно запрограммировать самим. Возможна, например, такая реализация:

```
for(MAX, MAX, J) -> [J(MAX)];
for(I, MAX, J) -> [J(I)|for(I+1, MAX, J)].
```

Код создает список, содержащий все числа, сформированные циклом **for**. Поместив вышеупомянутый код в файл **helloWorld.erl** — помните, что в Erlang весь код должен находиться в модулях, если вы не в оболочке — и экспортировав функцию **for**, можно использовать этот код так:

```
3> helloWorld:for(1,10,fun(I) -> 3 * I end).
[3,6,9,12,15,18,21,24,27,30]
4> helloWorld:for(-1,5,fun(I) -> I * I * I end).
[-1,0,1,8,27,64,125]
```

Как вы понимаете, выводом цикла **for** можно управлять динамически, меняя определение функции **fun(I)** — которая является параметром функции **for** — так же, как мы сделали бы в «обычном» цикле **for**.

Возможность создавать собственные управляющие конструкции — огромное преимущество, потому что вы не ограничены фиксированным количеством управляющих конструкций, как в процедурных и объектно-ориентированных языках программирования. Созданные вами управляющие конструкции также обычно лучше приспособлены для решения именно вашей проблемы и улучшают читаемость кода.


Пишем простую программу на Erlang

Код на Erlang для поиска чисел, принадлежащих последовательности Фибоначчи, таков (**linuxformat.erl**):

```
-module(linuxformat).
-export([fib/1]).
fib(N) when N == 0 -> 0;
fib(N) when N == 1 -> 1;
fib(N) when N == 2 -> 1;
fib(N) when N > 2 -> fib(N-1)+fib(N-2).
```

Объявление **fib/1** означает, что функция **fib()** принимает один аргумент. У функции **fib(N)** есть несколько ветвей, зависящих от значения параметра. Функцию **fib** можно запустить из оболочки Erlang следующим образом:

```
1> c(linuxformat).
```

Скорая помощь 

Стоит ли всегда пользоваться Erlang? Все задачи программирования Erlang эффективно решить не может, поэтому пользуйтесь и другими языками программирования, подходящими для решения конкретных задач. Для работы всегда следует выбирать самый удобный инструмент!

```
26 end.
27
28 process_line(Device, File) ->
29 case io:get_line(Device, "") of
30 eof -> [];
31 Line -> save_line(Line, File),
32 Line ++ process_line(Device, File)
33 end.
34
35 %% @doc This function takes two arguments.
36 %% It checks if an Integer (Line variable) is odd or even. If it is odd,
37 %% then it writes the Integer to a new file (File variable).
38 save_line(Line, File) ->
39 TrimLine = re:replace(Line, "(^\\s+)(\\s+$)", "", [global, {return, list}]),
40 %% io:format("TrimLine: ~s", [TrimLine]),
41 Number = list_to_integer(TrimLine),
42 if (Number rem 2) == 1 ->
43 io:format("odd\n"),
44 io:format(File, "~s~n", [TrimLine]);
45 true -> io:format("even\n")
46 end.
```

» Рис. 1. Полный исходный код модуля **processFile.erl**, включающий код для создания HTML-документации.

» Не хотите пропустить номер? Подпишитесь на www.linuxformat.ru/subscribe!

```
code$ erl
Erlang R15B01 (erts-5.9.1) [source] [64-bit] [async-threads:0] [kernel-poll:false]

Eshell V5.9.1 (abort with ^G)
1> cprof:start().
5614
2> processFile:readfile("data").
even
odd
odd
even
odd
odd
File closed.
"12\n23\n43\n32\n3\n21\n"
3> cprof:pause().
5731
4> cprof:analyse(processFile).
{processFile,14,
 [{{processFile,process_line,2},7},
 {{processFile,save_line,2},6},
 {{processFile,readfile,1},1}}]
5> cprof:stop().
5731
6>
```

➤ **Рис. 2. В Erlang есть собственные утилиты для профилирования. Здесь вы видите утилиту *cprof*, анализирующую профиль модуль Erlang.**

```
{ok,linuxformat}
2> linuxformat:fib(20).
6765
```

Ключевое слово **-module** в верхней части кода задает название модуля — **linuxformat**. Директива **-export** определяет полный список функций, которые будут видимы миру. Модуль **linuxformat** экспортирует единственную функцию **fib/1**. Заметьте, что указаны как имя функции, так и общее количество параметров функции. В Erlang две функции с одинаковыми именами, но с разным количеством параметров будут считаться совершенно разными функциями.

Подробнее о модулях Erlang

Модули в Erlang похожи на классы в объектно-ориентированных языках, а процессы — на объекты. У процессов в Erlang нет общей памяти (хорошо!), и они могут взаимодействовать только передачей сообщений. Процессы в Erlang не имеют ничего общего с процессами или потоками в UNIX.

После компиляции модуля в оболочке Erlang или в оболочке UNIX создается новый файл с таким же именем как у модуля, но с расширением **.beam**.

```
$ erl -compile linuxformat.erl
$ ll linuxformat.*
-rw-r--r-- 1 mtsouk mtsouk 620 Nov 13 09:32 linuxformat.beam
-rw-r--r-- 1 mtsouk mtsouk 159 Oct 13 20:05 linuxformat.erl
```

Ключевой компонент среды выполнения Erlang (Erlang Runtime System — ERTS) — процессор BEAM. BEAM — сокращение от “Bogdan’s Erlang Abstract Machine [Абстрактная машина Erlang Богдана]”. Это виртуальная машина, которая интерпретирует оптимизированный код BEAM. При компиляции программы на Erlang компилятор всегда конвертирует код в файл BEAM.

Работа с файлами

Перейдем к чтению и записи файлов в Erlang, воспользовавшись для этого модулем **processFile.erl** (рис. 1). Код прочитает указанный в качестве аргумента текстовый файл, содержащий числа, и запишет все нечетные целые числа из этого файла в другой файл под названием “LXFodd”. Встроенная функция **file:open(FileName, [read])** открывает файл на чтение. Чтобы открыть файл на запись, используйте следующий код:

```
{ok, FILE} = file:open(«output», write)
Самая важная и интересная часть модуля — следующая:
save_line(Line, File) ->
TrimLine = re:replace(Line, “(^\\s+)|(\\s+$)”, “”, [global,{return,list}]),
Number = list_to_integer(TrimLine),
```

```
if (Number rem 2) == 1 ->
io:format(“odd\n”),
io:format(File, “~s~n”, [TrimLine]);
true -> io:format(“even\n”)
end.
```

В Erlang нет оператора **return**. Все последовательные утверждения, принадлежащие к одному блоку, разделяются запятыми и завершаются точкой с запятой. Утверждение **if** должно охватывать все возможные варианты; потому и нужна строка **true -> io:format(«even\n»)**, хотя она не делает ничего полезного. Порядок различных операторов в **if** имеет значение, потому что выполняется только первый. С помощью регулярного выражения из переменной **Line** удаляются все пробелы, а также символ перевода строки. Переменная **TrimLine** также используется для записи нечетного целого числа в результирующий файл. Строка преобразуется в целое число с помощью выражения **Number = list_to_integer(TrimLine)**.

Программу можно скомпилировать и запустить из оболочки Erlang следующим образом:

```
33> c(processFile).
{ok,processFile}
34> processFile:readfile(“data”).
even
...
odd
File closed.
“12\n23\n43\n32\n3\n21\n”
```

По завершении всех операций записи не забывайте закрывать файл оператором **file:close(FILE)**.

Итак, подходит ли Erlang для системного программирования? Думаю, не очень, но вы все равно сможете сделать то, что хотите.

Профилирование и тестирование кода

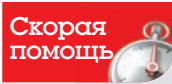
В Erlang есть три утилиты профилирования: *cprof*, *fprof* и *eprof*. *fprof* создает значительную нагрузку на систему, но позволяет профилировать огромные системы в лаборатории или окружении для симуляции. Две других утилиты подходят для профилирования рабочих систем, так как гораздо меньше нагружают систему (менее чем на 10 процентов).

Чтобы начать работать с *cprof*, можете запустить **cprof:start()**, приостановить ее, запустив **cprof:pause()**, и остановить, запустив **cprof:stop()**. Запустив *cprof*, нужно запустить программу, которую вы хотите профилировать, и затем приостановить *cprof*.

На рис. 2 показаны результаты профилирования модуля **processFile.erl**. Вывод показывает, что модуль выполнил в общей сложности 14 вызовов функций: один вызов **readfile()**, шесть — **save_line()** и семь — **process_line()**. Результаты вполне отражают реальность, так как **readfile()** вызывается всего один раз, а **process_line()** вызывается столько же раз, сколько **save_line()** плюс еще один раз, необходимый для определения конца файла.

Особенно полезно найти строки кода, которые не выполняются никогда — это явный признак того, что в коде что-то не так (ошибки!). Также, зная, какие функции выполняются чаще других, можно легко и уверенно понять, что стоит оптимизировать. Эту задачу можно выполнить с помощью анализатора Code Coverage, также предлагаемого Erlang. Проверка модуля **processFile.erl** анализатором Code Coverage включает следующие шаги:

```
2> cover:start().
{ok,<0.39.0>}
3> cover:compile(processFile).
{ok,processFile}
4> processFile:readfile(“data”).
```



При создании программ на Erlang разработчики помещают свой код в модули. Модули помогают легко и эффективно организовывать, размещать свой код и делиться им, и это — предпочтительный способ написания программ на Erlang.

Шесть правил

Архитектура Erlang следует шести правилам:

- » **Изоляция** Процессы в Erlang изолированы самой архитектурой. У каждого есть собственный стек и «куча», и для каждого отдельно собирается «мусор». Процессы также недоступна память других процессов, поэтому он не может негативно повлиять на них.
- » **Параллелизм** Процессы обладают параллелизмом. Поэтому теоретически все процессоры могут работать параллельно. Ныне, когда процессоры многоядерные, это очень хорошо, так

как процессы можно распределять по доступным ядрам.

- » **Определение ошибок** Процессы в Erlang могут определять ошибки. Также можно создать связь между двумя процессами. Следовательно, если процесс по какой-то причине дает сбой, об этом можно оповестить какой-то другой процесс. И если возникнет проблема, ее решение можно поручить кому-то другому.
- » **Идентификация ошибок** При сбое процесса сигнал ошибки содержит дополнительные данные,

предоставленные средой выполнения Erlang, которые помогут точно определить причину сбоя.

- » **Обновление ядра во время работы** Erlang можно изменять прямо во время работы, т.е. обновлять даже запущенные приложения!
- » **Надежное хранилище данных** Оно реализовано не в Erlang, а сторонними библиотеками. Для хранения данных можно воспользоваться *Mnesia*, *Riak* и другими продуктами. Доступ к базе данных есть у каждого процесса, т.к. эти данные разделяются между процессами Erlang.

```
...
5> cover:analyse_to_file(processFile).
{ok,»processFile.COVER.out»}
```

Результаты автоматически сохраняются в файле **processFile.COVER.out**, который надо будет распечатать или открыть в любимом текстовом редакторе.

Erlang и Mnesia

Mnesia — очень быстрая, распределенная СУБД, написанная на Erlang. Она входит в стандартный дистрибутив Erlang, и в ней можно хранить любые типы структур данных Erlang. В *Mnesia* таблицы можно хранить в оперативной памяти или на диске. В первом случае мы получим высокую скорость, во втором — сохранность [persistence].

Для создания новой базы данных *Mnesia* выполнит функцию **mnesia:create_schema()**. После запуска этой функции *Mnesia* создает новую структуру каталогов для размещения файлов базы данных. На рис. 3 показан полный пример использования *Mnesia* в Erlang с применением кода из модуля **myMnesia.erl**.

Команда **mnesia:create_schema(node())** инициализирует новую базу данных *Mnesia* только на текущем узле — из-за параметра **node()**.

Обратите внимание, что в результате запроса **SELECT** строки могут идти в любом порядке. Функция **traverse_table_and_show()** считывает все данные таблицы и выводит их на экран.

Функция **add_magazine** реализована следующим образом:

```
add_magazine(Number, Year, Month, Pages) ->
Row = #linuxformat{number=Number, year=Year, month=Month,
pages=Pages},
F = fun() ->
mnesia:write(Row)
end,
mnesia:transaction(F).
```

Первая команда определяет новую запись и сохраняет ее в переменной **Row**. Функция **transaction()** выполняет функциональный объект **F** как транзакцию. Это стандартный способ сохранения записей в *Mnesia*.

Функцию **myMnesia:initialize_database_once()** запускается только один раз. Затем нужно запустить базу данных *Mnesia* вручную командой **mnesia:start()**. Как видно из рис. 3, в текущей реализации данные не персистентны, то есть при выходе из оболочки Erlang все данные таблицы “linuxformat” будут потеряны!

Генерация документации Erlang

Erlang позволяет включать в разрабатываемый модуль метаданные для генерации документации. Код следует документировать всегда — это поможет и другим, и вам самим.

Дополнительные строки, которые вы введете в модуль **processFile** для документации кода, не повлияют на функциональность

```
Eshell V5.9.1 (abort with ^G)
1> c(myMnesia).
{ok,myMnesia}
2> myMnesia:initialize_database_once().
stopped
3>
=INFO REPORT==== 21-Nov-2014:00:39:45 ===
 application: mnesia
  exited: stopped
  type: temporary
4> mnesia:start().
ok
5> myMnesia:traverse_table_and_show(linuxformat).
[]
6> myMnesia:
add_magazine/4          initialize_database_once/0
module_info/0         module_info/1
traverse_table_and_show/1
7> myMnesia:add_magazine(194, 2014, Nov, 2011).
* 1: variable 'Nov' is unbound
8> myMnesia:add_magazine(194, 2014, "Nov", 2011).
{atomic,ok}
9> myMnesia:add_magazine(195, 2014, "Dec", 215).
{atomic,ok}
10> myMnesia:traverse_table_and_show(linuxformat).
{{linuxformat,195,2014,"Dec",215}}
{{linuxformat,194,2014,"Nov",2011}}
[]
11> q().
ok
12> code:ls -l Mnesia.nonode@nohost/
total 16
-rw-r--r-- 1 mtsouk mtsouk 150 Nov 21 00:39 DECISION_TAB.LOG
-rw-r--r-- 1 mtsouk mtsouk 89 Nov 21 00:41 LATEST.LOG
-rw-r--r-- 1 mtsouk mtsouk 6784 Nov 21 00:39 schema.DAT
code$ erl
```

» Рис. 3. Здесь показан небольшой пример взаимодействия с базой данных *Mnesia* на Erlang.

или скорость выполнения модуля; эти строки будут использоваться для генерации вывода в формате HTML.

Чтобы сгенерировать HTML-файлы и поместить их в каталоге **documentation**, выполните следующую команду из оболочки Erlang:


```
edoc:files(["processFile.erl"], [{"dir", "documentation"}]).
```

После этого в каталоге появятся следующие файлы:

```
$ ls -l documentation/
total 32
-rw-r--r-- 1 mtsouk mtsouk 40 Nov 20 19:46 edoc-info
-rw-r--r-- 1 mtsouk mtsouk 2109 Nov 20 19:46 erlang.png
-rw-r--r-- 1 mtsouk mtsouk 471 Nov 20 19:46 index.html
-rw-r--r-- 1 mtsouk mtsouk 424 Nov 20 19:46 modules-frame.html
-rw-r--r-- 1 mtsouk mtsouk 1070 Nov 20 19:46 overviewsummary.html
-rw-r--r-- 1 mtsouk mtsouk 327 Nov 20 19:46 packagesframe.html
-rw-r--r-- 1 mtsouk mtsouk 2512 Nov 20 19:46 processFile.html
-rw-r--r-- 1 mtsouk mtsouk 895 Nov 20 19:46 stylesheet.css
```

Помните, что документация формируется только для функций, перечисленных в списке для экспортирования, а для остальных функций не формируется, даже если вы написали для них документацию (@doc).

Как однажды сказал Стив Джобс: «Процессорная индустрия добавляет в процессоры новые и новые ядра, но как программировать для них, никто не знает. Два ядра — ОК, четыре — не уверен, восемь — забудьте об этом». Erlang же по своей архитектуре поддерживает процессоры с несколькими ядрами!

Чем больше вы будете практиковаться, тем эффективнее станет ваш код. Хорошая книга об Erlang — *Programming Erlang, 2nd Edition* [Программирование на Erlang, 2-е издание] Джо Армстронга [Joe Armstrong]. 

Скорая помощь

Чтобы ваш код был устойчивым к ошибкам, пишите его с использованием изолированных компонентов. Такие компоненты могут работать параллельно. Соединив эти две идеи вместе, можно прийти к выводу: то, что изолировано и выполняется параллельно, поддается масштабированию.

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 Синхронизацию с Google Drive
- 2 Копирование DVD Linux Format
- 3 Простое решение для удаленного доступа
- 4 Медленное копирование с scp
- 5 Утерянный пароль для входа в систему
- 6 SD-карты большой емкости

1 Google Drive в Linux

У меня есть более 100 ГБ свободного места на Google Drive (оно дается с большинством хромбуков), но единственный способ воспользоваться им в Linux — загружать файлы через web-браузер. А я хотел бы синхронизировать каталоги автоматически в фоновом режиме или с помощью *cron*. Возможно ли это?

Эдвин Берри [Edwin Berry]

Как это ни печально, но Google предпочел не выпускать Linux-клиент для Google Drive. Однако альтернативы есть. Одна из них — *gsync* (<http://bit.ly/Gsync4GoogleDrive>), которая эмулирует параметры *rsync*, так что Вам не придется изучать ничего нового. Это программа на Python, которую лучше всего установить с помощью *pip*, менеджера пакетов Python. Если раньше Вы не пользовались *pip*, возможно, придется его установить. Сначала убедитесь, что у Вас установлен пакет *python-setuptools* (в большинстве дистрибутивов он включен по умолчанию), затем запустите команду

```
sudo easy_install pip
```

Теперь можно установить *gsync*, командой

```
sudo pip install gsync
```

Pip позаботится обо всех зависимостях, которые нужно разрешить для установки *gsync*. Перед тем, как попробовать *gsync*, нужно исправить одну ошибку, которая иногда препятствует загрузке файлов. Чтобы ее исправить, откройте от имени root файл `/usr/lib64/python2.7/site-packages/libgsync/drive/__init__.py` в текстовом редакторе и перейдите к строке 744. Там Вы найдете следующий фрагмент кода:

```
for k, v in properties.iteritems():
```

```
    body[k] = _Drive.utf8(v)
```

Замените его на

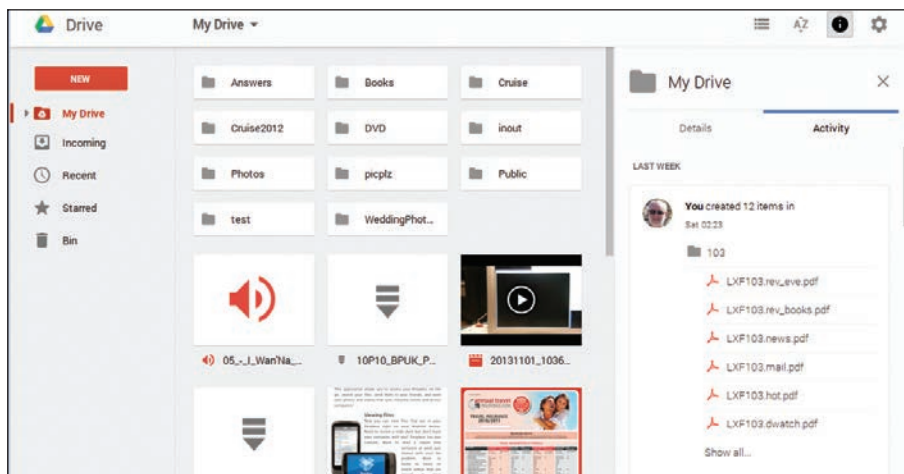
```
for k, v in properties.iteritems():
```

```
    if v is not None:
```

```
        body[k] = _Drive.utf8(v)
```

Отступы (как и всегда в Python) имеют значение. Теперь один раз запустите команду

```
sudo gsync
```



Google не предоставляет Linux-клиента для Google Drive, но *gsync* сделает эту работу.

которая выдаст краткое сообщение об использовании команды. Ее нужно запустить однократно и с правами *root*, чтобы скомпилировать только что сделанные изменения. Теперь Вы можете синхронизировать локальный каталог с каталогом на Google Drive, командой

```
rsync -rv /local/directory drive://directory
```

При первом запуске команды *gsync* выведет длинный URL. Откройте его в браузере, и Google попросит Вас войти в свою учетную запись, если Вы еще не вошли, а затем спросит у Вас разрешение на доступ к учетной записи. Когда Вы подтвердите разрешение, Google сообщит Вам код, который нужно скопировать и вставить в терминал, после чего *gsync* начнет копировать файл в учетную запись на Google Drive. Процесс аутентификации нужно пройти всего один раз, и *gsync* сохранит токен для автоматической аутентификации в будущем.

Теперь *gsync* в порядке, и можно запустить команду

```
gsync --help
```

чтобы увидеть список доступных параметров. Параметры соответствуют аналогичным параметрам *rsync*: среди них есть все, которые Вам нужны, и их синтаксис идентичен. Теперь можно синхронизировать данные на Google Drive или с него по желанию либо добавив соответствующую команду в свой *crontab*.

2 «Большие» DVD

В библиотеке Олдриджа хранится подборка всех номеров журнала *Linux Format*, которые используются любителями открытого ПО здесь в Бэрре, штат Вермонт, США. Диски, которые идут с номерами, мы не распространяем,

но копируем, и эти копии предоставляются нашим читателям.

Недавно при копировании нескольких дисков у нас возникла проблема. Кажется, проблема в объеме диска, который превышает максимальный размер DVD+R. Можете ли вы помочь нам скопировать эти диски, чтобы пустить их в обращение?

Пол Флинт [Paul Flint]

Прежде всего, нам очень приятно слышать об организациях, подобных Вашей, которые делают так много для поддержки открытого сообщества. Ваша проблема связана с тем, что Вы пользуетесь DVD+R. Формат DVD+R содержит больше данных для коррекции ошибок, чем DVD-R, и поэтому его объем чуть меньше. На DVD-R объемом 4,7 ГБ примерно на 7 МБ больше места, чем на DVD+R, и его объем ближе к объему штампованного DVD. Мы стараемся максимально заполнять диски, и на диске, который я делаю сейчас, всего 1 МБ свободного места, поэтому такие диски просто не копируются на диски чуть меньшего объема.

Все современные приводы поддерживают оба формата, поэтому решение простое — воспользуйтесь дисками DVD-R. Хотя наши диски штампуются, а не записываются, для тестирования мы используем DVD-R, чтобы убедиться, что диски будут работать при записи на этот носитель.

3 Простой удаленный доступ

Родственники часто просят меня помочь с компьютером под Windows. Они неужели работают с компьютером, и им часто приходится помогать; но к тому же они живут

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 данные команды без предшествующей `sudo`.

очень далеко. Я знаю, что мог бы удаленно подключиться к их компьютеру через VNC, но не думаю, что это позволит разговаривать с ними во время настройки компьютера. Есть ли более простой метод, который одинаково хорошо работал бы на Linux и на Windows?

Под Диггз (Rod Diggs)

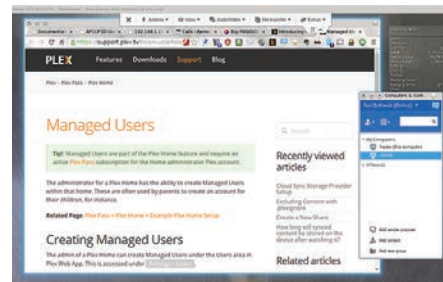
Остановка и настройка VNC бывает довольно сложной, особенно если Вам нужно обращаться к нему удаленно. Вам как минимум придется открыть и перенаправить необходимые порты на роутере, но открывать VNC-сервер Гадкому Большому Интернету — не лучшая

идея. А значит, в идеальном случае нужно настроить и VPN, так что эта задача довольно затратная.

Существует несколько вариантов, и все они просты, но ни один из них не является открытым. Самый свободный — пожалуй, *Chrome Remote Desktop* от Google, но он включает установку *Chrome*, если у Вас его еще нет, затем расширения для *Chrome*, да еще серверного компонента. Наверное, самый простой и поэтому самый подходящий для Вас вариант — это *TeamViewer*. Это коммерческий продукт с солидным ценником, но для некоммерческого использования он бесплатен. На компьютере с Linux его можно установить с помощью менеджера пакетов дистрибутива. Все, что должен сделать Ваш родич — зайти на <http://teamviewer.com>, нажать кнопку Download [Загрузить] и выполнить установку обычным образом. После запуска мастер настройки неконтролируемого доступа можно просто пропустить, и откроется главное окно *TeamViewer* с идентификатором и паролем. Попросите родственника прочесть их, введите их в свою копию *TeamViewer*, запущенную локально, и Вы подключитесь к удаленному рабочему столу.

Чтобы каждый раз не обмениваться паролями по телефону, создайте учетную запись *TeamViewer* на своем компьютере и зайдите в него с удаленного компьютера. Потом Вы сможете указать для компьютера имя и пароль. Повторите эти действия для всех компьютеров, которые хотите добавить. Теперь в Вашем *TeamViewer* отображается список компьютеров, связанных с Вашей учетной записью, и их статус в Сети (онлайн они или нет).

Возможно, Вы заметили, что клиент для Linux выглядит странновато. Дело в том, что на самом деле это программа для Windows с библиотеками *Wine* для запуска в Linux. Пусть это выглядит халтуркой, но если в итоге означает, что пакет для Linux всё-таки выпустили, оно и неплохо. *TeamViewer* по-настоящему кроссплатформенный, есть даже клиент для Android. Этот клиент однонаправленный — Вы можете управлять компьютером с мобильного устройства, но не наоборот.



» *TeamViewer* не является открытым, но это один из самых простых способов управления удаленным компьютером для обычного пользователя.

Как и с любой коммерческой программой, которая в бесплатном варианте ограничена (хотя в данном случае ограничение невелико), всегда есть риск того, что она перестанет быть бесплатной, как в прошлом году сделала Logmein; поэтому всегда стоит думать и об альтернативах, просто на всякий случай.

4 Медленная scp

В Я часто копирую файлы по сети с помощью *scp*. Она легкая и удобная, не нужно настраивать сетевые ресурсы и тому подобное. Но она не такая быстрая, что очень заметно при перемещении больших файлов (свыше 1 Гб). Скорости *scp* значительно меньше, чем при копировании на ресурс *Samba* или *NFS*. Это непреодолимое ограничение *scp* или ее можно как-то ускорить?

Дэйв Питмен (Dave Pitman)

О Падение скорости, которое Вы наблюдаете, происходит из буквы "s" в *scp* — «безопасный [secure]». SSH, механизм, используемый *scp* для передачи файлов, шифрует все данные перед отправкой, а при приеме на другом конце они расшифровываются. Если окажется, что в департаменте центрального процессора это не слишком приветствуют, то Вы увидите падение производительности, особенно »



Коротко про...

Потребление памяти

Индикатор объема памяти на рабочем столе показывает, что через несколько часов после начала работы системы осталось всего несколько килобайт? Может показаться, что нужно добавить еще памяти, но это не так. Это не ошибка — ядро просто использует память наилучшим образом.

Хотя бесполезное злоупотребление памятью — это трата ресурсов, простаивание памяти — еще худшая трата, поэтому ядро Linux выделяет неиспользуемую память на такие вещи, как кэши файловой системы, ради повышения

производительности. Это можно увидеть в настройках буферов в выводе команды `top`:

```
Mem: 1028092k total, 1018704k used, 9388k free, 49760k buffers
```

Данная строка показывает, что из 1 Гб оперативной памяти на компьютере свободно менее 10 МБ.

Из используемой памяти примерно половина используется в качестве буферов. Они значительно ускоряют доступ к диску, кэшируя записанные данные в память и считывая больше, чем вам необходимо, в упреждение следующей

операции чтения. Если эта память необходима для чего-то более важного, например, данных программы, она незамедлительно освобождается. В памяти остаются не только данные файловой системы: попробуйте проверить объем свободной памяти, затем запустить и закрыть большую программу, такую как *Firefox*, и вы увидите, что объем свободной памяти уменьшился. Теперь снова запустите *Firefox* и посмотрите, насколько быстрее он стал работать. Так происходит потому, что ядро хранит данные в памяти готовыми для следующего употребления. То же относится к разделяемым библиотекам, поэтому с программами из одного рабочего стола получается работать быстрее, чем со смесью программ из Gnome и KDE.


```
IW /etc/shadow Row 24 Col 1 1:02 Ctrl-K H for help
portage:!!:0:0:99999:7:::
nobody:!!:9797:0:::
ldap:!!:0:0:99999:7:::8272504
sshd:!!:13622:0:99999:7:::
cron:!!:13648:0:99999:7:::
rpc:!!:13648:0:99999:7:::
messagebus:!!:16087:::
mysql:!!:13649:0:99999:7:::
haldaemon:!!:13649:0:99999:7:::
nelz:$6$хqTnzCbkut$P.0AF.piU7Jxno.4eCty/vuNa5/TclgsuUTa4.RIUH0Ml
backup:!!:0:0:99999:7:::0
mythtv:!!:0:0:99999:7:::0
scponly:!!:0:0:99999:7:::0
apache:!!:13649:0:99999:7:::
gkrellmd:!!:13649:0:99999:7:::
clamav:!!:13649:0:99999:7:::
ntop:!!:13649:0:99999:7:::
games:!!:13649:0:99999:7:::
at:!!:13649:0:99999:7:::
```

➤ Если больше ничего не помогает, то для сброса пароля можно изменить `/etc/shadow` — но сохраните копию, чтобы не потерять доступ к компьютеру из-за случайной опечатки.

у отправителя. Это влияет не только на передачу файлов, но и на производительность компьютера в целом, которая снижается, когда процессор занят шифрованием файлов на несколько гигабайт.

Безопасность особенно приветствуется при использовании SSH для открытого интернет-подключения, особенно потому, что в этом случае скорость, вероятно, все равно будет ограничивающим фактором. Хорошие новости в том, что как и большинство других аспектов SSH, шифрование можно настроить в соответствии со своими потребностями. У SSH есть несколько шифраторов, и используемый по умолчанию `3des` считается достаточно надежным для открытых подключений. Если Вы передаете файлы по локальной сети и Вам не нужна повышенная безопасность, можете воспользоваться более слабым, но более быстрым шифратором. Например, команда

```
scp -c arcfour,blowfish somefile user@somehost:/somedir
```

попыtet зашифровать поток с помощью шифратора `arcfour`, старого и потенциально слабого, но зато очень быстрого. Если это не получится из-за того, что один из компьютеров не поддерживает данный шифратор, `scp` попыtet `blowfish`, еще один быстрый шифратор. Если выдается ошибка “no matching cipher found [не найдено соответствующего шифратора]”, убедитесь, что используемые шифраторы указаны в строке `ciphers` файла `/etc/ssh/ssh_config` на обоих компьютерах.

Пока все хорошо, но теперь приходится набирать более длинную команду, и Вы теряете время, сэкономленное на быстром выполнении команды. Можно было бы воспользоваться алиасом, но тогда более слабый шифратор будет применяться и на нелокальных подключениях. Ответ лежит в определении хоста в файле настройки SSH. Глобальный конфигурационный файл — `/etc/ssh/ssh_config`, а настройки для отдельных пользователей, имеющие приоритет над любыми глобальными, задаются в `~/.ssh/config`. Строки

```
Host a.local.computer
Ciphers arcfour
```

сообщают SSH, что применять шифратор `arcfour` нужно только для подключений к этому компьютеру. В строке `host` допускаются маски и шаблоны, и каждый компьютер не нужно специфицировать отдельно, например:

```
Host a.local.computer 192.168.0.* *.my.domain
Ciphers arcfour
```

Если хотя бы одно из условий верно, шифратор будет заменен. Так же можно изменить и другие настройки. Применяются все настройки после строки `Host` до появления следующей строки `Host`. Также можно изменить настройки по умолчанию для всех остальных хостов с помощью строки:

```
Host *
settings
```

Используется первая найденная директива `Host`; если Вы используете глобальную директиву, поместите ее в конец файла.

5 Маленькая большая карта

В Я купил карту MicroSDXC на 64 Гб — они сейчас так подешевели! — но эта карта не читается в Linux. Я думал, что на всех SD-картах применяется файловая система MS-DOS FAT, которая в Linux поддерживается. Что нужно сделать, чтобы воспользоваться этой картой? Если она повреждена, нужно ли ее реформатировать?

Том Олсон [Tom Olson]

О Стандарты SD-карт определяют файловую систему по умолчанию, которая должна использоваться на карте, а также более очевидные аппаратные аспекты. SD распространялась на карты до 4 Гб, SDHC появилась для карт от 4 до 32 Гб, и теперь у нас есть SDXC для карт от 64 Гб до 2 Тб (хотя, думаю, в наши телефоны такие попадут не скоро). Карты MicroSD, хотя и физически отличаются от обычных SD-карт, в отношении файловой системы следуют тем же спецификациям. Стандартная файловая система для карт SD и SDHC — на самом деле FAT, поэтому они и работают практически везде. Однако у FAT есть

много ограничений, и одно из них — максимальный размер файла в 4 Гб. Поэтому спецификация SDXC принудительно заставляет использовать exFAT, которая лучше подходит для больших файлов, но является проприетарной и лицензируется Microsoft. Кстати, именно поэтому мы часто видим устройства, которые поддерживают карточки только до 32 Гб. Проблема не в том, что им не прочтеть карточки на 64 Гб, а в том, что им не прочтеть файловую систему, так как они не имеют соответствующей лицензии.

В зависимости от того, как Вы собираетесь пользоваться карточкой, у Вас есть несколько вариантов действий. Первый вариант — реформатировать карту с файловой системой FAT32. Максимальный размер файла будет ограничен 4 Гб, но совместимость будет максимальной. Именно так я поступил с картой объемом 64 Гб в планшете Hudl2. FAT медленнее, чем exFAT, но здесь ограничивающим фактором скорее всего будет железо.

Другой вариант — установить на компьютер с Linux драйвер exFAT. Samsung опубликовала исходный код для драйвера exFAT в ядре, лицензированный GPL, но проблемы с патентами помешали включить его в ядро Linux. Это не остановило активное сообщество, и оно создало драйвер с помощью FUSE. Чтобы им воспользоваться, нужно установить два пакета, обычно это `exfat-fuse` и `exfat-utils`. Первый — драйвер FUSE, а второй содержит пользовательские утилиты, версии `mkfs`, `fsck` и `mount` для exFAT.

Теперь карта должна автоматически распознаться после того, как Вы вставите ее в компьютер. Если этого не произошло, может потребоваться загрузить модуль FUSE вручную:

```
sudo modprobe fuse
```

Если после этого файловая система распознается, можно сделать так, чтобы данный модуль всегда загружался при загрузке системы. Для этого от имени суперпользователя-root создайте файл `/etc/modules-load.d/fuse.conf`, содержащий единственную строку “fuse”. Проще всего это сделать командой

```
echo fuse | sudo tee -a /etc/modules-load.d/fuse.conf
```

Учтите: этот каталог есть не во всех дистрибутивах, он может быть и другим, например, `/etc/modules`. Если каталог `/etc/modules-load.d` существует, воспользуйтесь им.

6 Утерянный пароль

В Недавно я решил поработать на компьютере, который давно не включал, и понял, что забыл пароль своего пользователя. Не можете ли мне восстановить пароль?

Коллин Робинсон [Colin Robinson]

О Восстановить прежний пароль невозможно, но можно задать новый пароль, который Вы, надеюсь, не забудете. Как это сделать, зависит от дистрибутива и других аспектов системы, о которых Вы не сказали. Поэтому вот возможные варианты, начиная с самого простого и заканчивая тем, который точно сработает.

Есть ли у Вас root-доступ к компьютеру? Если да, достаточно запустить команду

```
passwd username
```

Она напомнит Вам пароль пользователя, с которым затем Вы сможете зайти в систему. Если это не подходит, следующий вариант — попробовать установочный диск дистрибутива. Есть ли у него опция аварийной загрузки, которая загрузит Вас в оболочку? Это не то же самое, что опция восстановления в меню загрузки Ubuntu, поскольку немедленно дает Вам root-доступ. Если получилось, можете выполнить приведенную выше команду. Если специального пункта аварийной загрузки нет, загрузите систему как обычно с жесткого диска. В меню загрузки выберите пункт обычной загрузки своего дистрибутива и нажмите e, чтобы отредактировать его. Если в нем есть раздел `init=`, удалите его и добавьте следующий:

```
init=/bin/sh
```

`init` — первая программа, запускаемая ядром, и она запускает все остальное. Эта строка говорит, что вместо всего остального нужно запустить оболочку. Теперь запускайте команду `passwd` как описывалось. Если Вы получаете ошибку «файловая система доступна только для чтения», она связана с тем, что ядро обычно монтирует корневую файловую систему таким образом, и наше изменение `init` не позволило дойти до того момента, когда она монтируется для чтения/записи. Чтобы этого исправить, выполните команду

```
mount / -o remount,rw
```

Теперь запустите `passwd` и перезагрузитесь. Если ничего из вышеперечисленного не помогло,

остается перейти к решительным действиям. Загрузитесь с Live CD, например, с System Rescue CD, и смонтируйте корневую файловую систему. Например, если Ваша корневая система располагается на `/dev/sda3`, выполните команду

```
mount /dev/sda3 /mnt/custom
```

Если у Вас есть сомнения насчет того, какая файловая система является корневой, попробуйте разные; Вам нужна та, что содержит `bin`, `etc`, `lib`, `usr` и т.д. Теперь измените файл `shadow` командами

```
cp -p /mnt/custom/etc/shadow /mnt/custom/etc/shadow.bak
```

```
nano /mnt/custom/etc/shadow
```

```
chmod 640 /mnt/custom/etc/shadow
```

Первая команда делает копию файла на случай, если Вы совершите ошибку. Вторая — загружает файл в редактор *nano*. Каждая строка файла `shadow` содержит девять полей, разделенных двоеточиями. Первое поле — имя файла, а второй — зашифрованный пароль. Удалите все между первым и вторым двоеточиями в строке для своего пользователя. Затем нажмите `Ctrl+X` для сохранения файла и выхода из редактора. Третья команда делает так, что файл не смогут читать обычные пользователи: это требование безопасности.

После выполнения этих действий Ваш пользователь останется без пароля. Можете перезагрузиться и зайти в систему как свой пользователь, не забыв, что первым делом нужно будет задать новый пароль. **LXF**

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://sourceforge.net/hardinfo.berlios>) — она сохранил 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
```



Часто задаваемые вопросы

Разделы и файловые системы

» Зачем в Linux столько всяких разделов и файловых систем?

Жесткий диск делится на разделы — отдельные секции, которые выглядят как диски. Для хранения данных на каждом разделе используется файловая система.

» Почему в Linux несколько разделов? В Windows всего один!

Изолированное хранение данных различных типов может повысить производительность и безопасность. Данные раздела подкачки `swap` более эффективно размещать в отдельной файловой системе, чем в файле в каталоге файловой системы раздела `root`. Неконтролируемый процесс может испортить данные `swap`, но не затронет ничего другого.

» И разделов `swap` и `root`, как в Ubuntu, будет достаточно?

Да, достаточно, но не оптимально. В настольной системе (включая ноутбуки) по меньшей мере каталог `home` должен размещаться на отдельном разделе. Тогда ваши персональные настройки и данные будут отделены от файлов ОС.

» Зачем мне это нужно?

Однажды вам захочется попробовать другой дистрибутив или установить более свежую версию текущего. Если все данные находятся в разделе `root`, все они будут уничтожены во время установки, и перед началом установки вам придется копировать все свои настройки и файлы на внешний диск. Если `home` находится на отдельном разделе, то приличный установщик оставит их нетронутыми. После установки можно загрузить компьютер и начать пользоваться новой системой точно так же, как прежней.

» Есть еще какие-то разделы, которые у меня должны быть?

Отдельный раздел для каталога `/var` для серверов — определенно хорошая идея. Это также удобно для настольных систем, поскольку при активном заполнении лог-файлов, хранящихся в `/var/log`, место на системном разделе не закончится.

» Типов файловых систем и так много — зачем нам `ext2`, `ext3`, `reiserfs`, `XFS`, `jfs` и все прочее?

У каждой файловой системы есть свои сильные и слабые стороны. `Ext2` — первоначальная файловая система Linux, и она работает очень быстро, но не слишком хорошо справляется с внезапным отключением системы. `Ext3`, как и остальные файловые системы, журналируемая. Это означает, что в случае падения системы или отключения питания вы сможете восстановить

данные. Кроме того, она быстрее и отличается большей надежностью за счет небольшого снижения скорости. `Reiserfs` особенно хорошо работает с маленькими файлами и с точки зрения использования дискового пространства, и с точки зрения скорости. `XFS` исключительно быстро работает с большими файлами, но плохо справляется с отключением питания, поэтому ее лучше использовать там, где имеется поддержка батарей. `Ext4` сочетает множество преимуществ других файловых систем, а `Btrfs` — новая, но многообещающая файловая система.

» Какая из файловых систем самая лучшая?

Точно так же можно спросить: «Что лучше, *Vi* или *Emacs*, либо KDE или Gnome?», поэтому я не буду отвечать на этот вопрос, спасибо.



LXF Hot Picks



Александр Толстой

Надевает защитные очки и огнеупорные перчатки, чтобы взяться за самые горячие свободные программы из Сети — с пылу, с жару.

Fifth » XFDesktop » GNU Hello » QuiteRSS » Audex » XDecorations » Debreate » Chocolate Doom » VoltAir » Budgie Desktop » ClamAV

Web-браузер

Fifth

Версия: 0.1.1 Сайт: <http://fifth-browser.sourceforge.net>

Честно говоря, мы не ожидали, что уход Opera 12.x несколько лет назад вызовет столь бурные протесты и даст такой толчок его последователям с открытым кодом. Мы уже писали о браузере *Otter* [HotPicks, LXF191/192], а на сей раз представляем вам другой браузер, продолжающий «классическую» философию Opera, под названием *Fifth*.

Fifth основан на наборе инструментов FLTK и движке Webkit, что является редкой комбинацией. Он заявляет, что «работает со скоростью света», и, похоже, это правда, потому что *Fifth* загружается меньше чем за секунду даже на нашем старом оборудовании. Хотя версия

0.1.x — только основа для более богатого функциями браузера, сайт *Fifth* обещает просто потрясающие вещи. Основная идея проекта — минимализм. *Fifth* не поддерживает Flash и WebGL, и не содержит никаких расширений JavaScript. Благодаря *Fifth*, ваши действия в Сети — под полным вашим контролем: *Fifth* не позволяет сайтам отслеживать вас, намеренно предоставляя неверную информацию почти

«Для тех, кто озабочен сохранением своей конфиденциальности.»



» Fifth делает вас неразличимыми в толпе, не позволяя вас отследить.

по всем пунктам, по которым можно вас вычислить.

При всём при том, работа с *Fifth* исключительно позитивна. Браузер невероятно быстр и имеет очень аккуратный и незахламленный интерфейс. Он напоминает ранние версии *Opera*, и предлагает только самые основные функции: вкладки, закладки, журнал, менеджер загрузок, сохранение пароля, быстрый набор и еще несколько. Сильная сторона *Fifth* — функция Page settings, которая позволяет сохранять предварительные индивидуальные настройки (CSS, Javascript и Images switches) для разных сайтов.

В нашем тесте HTML5 *Fifth* набрал скромные 345 очков, но был определен там как «неизвестный браузер», чем разработчики *Fifth* вправе гордиться. На данный момент все его функции реализованы в родном коде C/C++, что помогает ему поддерживать высокую производительность и очень низкое потребление памяти. Например, сам браузер без каких-либо загруженных страниц (по умолчанию он запускается со скоростью набора) занимает всего 3,2 МБ. *Fifth* может стать отличным решением для машин с невысокой производительностью, вместе с другими программами на FLTK. Кроме того, это истинная находка для тех, кто озабочен сохранением своей конфиденциальности и желает избежать отслеживания и учета. Из *Fifth* они могут быть практически невидимы для аналитиков.

Исследуем интерфейс Fifth

Знакомое управление

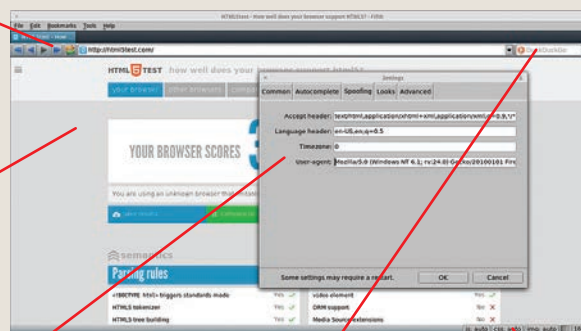
У *Fifth* весьма классическая внешность: сочетание простой графики и кнопок FLTK с симпатичными значками Crystal.

Рендеринг Webkit

Fifth делает солидную работу по рендерингу страниц с помощью WebKit. Если ваши любимые сайты не зависают от сторонних плагинов, то вам вполне должно хватить того, что здесь есть.

Соккрытие вашей истинной личности

Настройки User Agent предназначены для сокрытия вас от всевидящих глаз Большого Брата. Кем же мы будем сегодня? Может быть, пользователем Firefox на Windows?



Утки на страже анонимности

Вы получите правильные, но не персонализированные результаты поиска благодаря выбору по умолчанию — DuckDuckGo [duck — *англ.* утка]. Google тоже здесь доступен, в выпадающем меню.

Быстрое управление

Эти крошечные переключатели позволяют вам мгновенно принять решение о том, хотите ли вы, чтобы на текущем сайте работали JavaScript и CSS или отображались изображения.

Менеджер рабочего стола

Xfdesktop

Версия: 4.10.3 Сайт: <http://bit.ly/Xfdesktop>

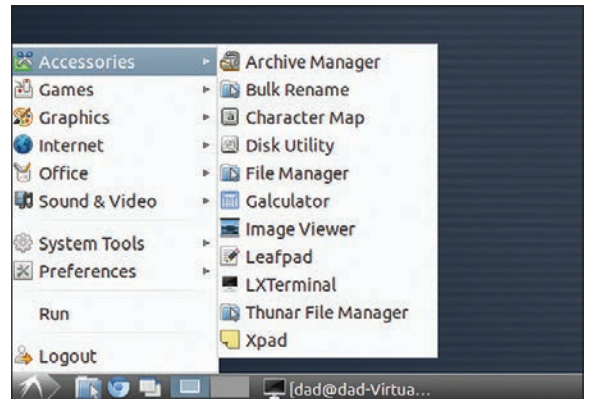
Название и номер версии уже говорят нам о том, с чем нам придется иметь дело: после длительной паузы разработчики *Xfce* наконец разродились новым стабильным релизом *Xfdesktop* — менеджера рабочего стола среды *Xfce*.

Сам по себе *Xfce* известен как традиционный и легковесный рабочий стол с простыми в использовании, но полнофункциональными элементами. Одним из его недостатков в последнее время была крайне медленная разработка, и даже некоторые преданные пользователи успели решить, что он заброшен. Но это оказалось не так, и новый релиз, хотя и не основной, доказывает, что проект *Xfce* скорее жив. Кое-кто считает, что его разработка в застое, но для многих это как раз преимущество, а не проблема, потому что *Xfce* реализует очень стабильную и сильную парадигму того, что люди хотят видеть на своем рабочем столе и что они хотят с ним делать.

Xfdesktop — один из ключевых компонентов среды *Xfce*. Он работает со значками рабочего стола, контекстным меню и настройками рабочего стола. Новый релиз предлагает ряд усовершенствований — например, меньшее время ожидания при изменении размера значков, счетчиком, опцию проверки эскизов на новом месте, отладки в коде загрузки подменю шаблона, более стабильную работу корзины и многое другое.

Исправленный пакет *Xfdesktop* — долгожданное обновление для всех пользователей *Xfce*; он по-прежнему использует ветвь 4.10. Взглянув на дорожную карту проекта *Xfce* (<http://bit.ly/1xOkRZW>), вы обнаружите, что выход версии 4.12 планировался на март 2013, но релиз пока что

«Долгожданное обновление для всех пользователей Xfce.»



► *Xfdesktop* теперь внутри белый и пушистый, благодаря обновленному и отлаженному коду.

ожидается. И хотя дата выхода 4.12 остается неизвестной, любое обновление *Xfce* — это не только признак того, что проект жив, но и долгожданные отладки, и новые функции. Мы заметили, что GitHub проекта весьма активен, и обновления на нем происходят весьма часто.

Помимо отладки мелких неприятностей, *Xfdesktop 4.10.3* также предлагает обновленный перевод, затронувший десятки поддерживаемых языков. Если вы используете Xubuntu или Manjaro (в обоих по умолчанию предусмотрен *Xfce*), мы бы предложили вам проверить наличие обновлений.

Программа приветствия GNU

GNU Hello

Версия: 2.10 Сайт: <http://ftpmirror.gnu.org/hello>

Это небольшое приложение-пример используется как справочник по структуре, документации, структуре релиза и прочим аспектам типового ПО GNU. Все это помогает новичкам в мире GNU узнать его и улучшить свои навыки правильной работы с программами.

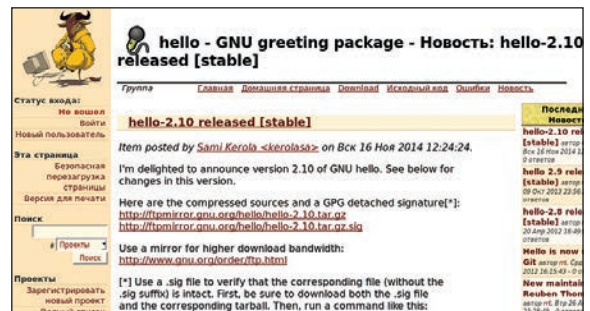
Программа называется *Hello* — это дань уважения знаковому примеру “Hello, world!”, который большинство использует в качестве первого опыта при изучении нового языка программирования и который впервые был использован во внутреннем документе Bell Laboratories, написанном Брайаном Керниганом [Brian Kernighan] по программированию на C. В некий момент народ в проектах GNU осознал, что непрограммирующие люди не имеют шансов обзавестись в своей оболочке каноническим приветствием “Hello, world!”, и решил это исправить.

Утилита *GNU Hello* является частью каждого крупного и большинства мелких

дистрибутивов Linux. Просто введите **hello**, и получите в ответ “Hello, world!” в переводе на ваш язык, который берется из локальной системной папки. Введите **hello -g <your_text>**, чтобы заменить текст по умолчанию на свой. И, наконец, используйте **hello -t** для получения приветствия по умолчанию, но без восклицательного знака (поэтому оно и называется традиционным).

Готовый *GNU Hello*, поставляемый с вашим дистрибутивом, мы брать не будем, а скомпилируем утилиту из исходника. Для тех, кто раньше не компилировал приложений Linux, это отличный способ начать. У *GNU Hello* очень понятная структура и никаких дополнительных зависимостей.

«./configure для GNU Hello выполняется дольше, чем make.»



► GNU-бвана уведомляет об обновлении: не забудьте выбрать ближайшее к вам зеркало, потому что файл весит аж 725 КБ!

Она использует *Gettext* и некоторые другие основные компоненты для разработчиков. Также весьма забавно, что команда **./configure** для GNU Hello выполняется дольше, чем обычная команда **make**.

Новая версия утилиты 2.10 содержит обновления перевода, более точные правила для компиляции map-страницы (**man hello**) и различные обновления для обеспечения отличной поддержки пакета *Autotools*.

Есть также некоторые другие улучшения, о которых стоит упомянуть: теперь основное приветствие “Hello, world!” берется из файлов **.po**, а функцию **fprintf(stderr)** заменила **errgr()**, которая является частью *Glibc*.

Читалка RSS

QuiteRSS

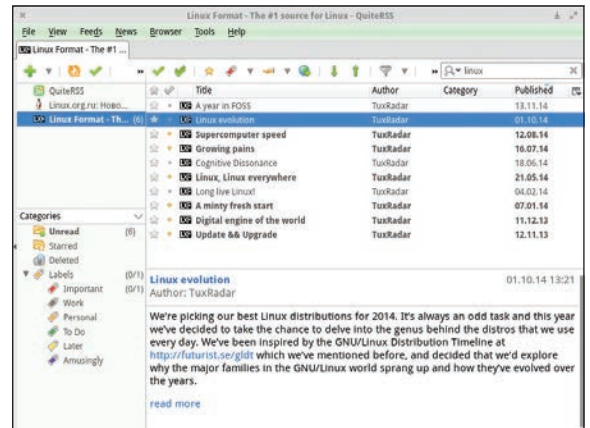
Версия: 0.17.1 Сайт: <http://quiterss.org>

Чем снова по бесконечным сайтам с новостями Linux, лучше завести специальную программу для сбора этих новостей. С 1995 г. RSS постоянно доказывает свое удобство в качестве метода получения новостей и статей. В самом начале эта аббревиатура означала Rich Site Summary, а не Really Simple Syndication, как теперь. Многие сайты создают собственные ленты RSS, разрабатываемые как отдельные XML-страницы на специальных URL. Содержание ленты RSS можно просматривать в любом веб-браузере, но в первую очередь она предназначена для использования с клиентами RSS, которых насчитывается немало, как сетевых, так и настольных; и *QuiteRSS* — отличный выбор настольного приложения.

В *QuiteRSS* множество функций, но они не лезут вам на глаза, если, конечно, вы сами того не захотите. Верхняя левая панель показывает список лент и панель инструментов для добавления новых лент или для изменения уже имеющихся.

Прямо под ней — дерево с цветными ярлыками, которые позволяют маркировать ленты и легко их находить. Основная часть окна похожа на почтовый клиент, и отображает список новостей из лент и содержание выбранной новости. Нижняя панель, представляющая содержание, является полнофункциональным веб-браузером — вы в этом убедитесь, как только щелкнете по любой ссылке внутри новости. Здесь имеется расширение AdBlock, и оно позволяет переключать отображение изображений, но не позволяет перейти на индивидуальный веб-адрес в строке ввода. Вместо этого вы видите кнопку «глобус» на панели инструментов, чтобы открыть URL в вашем браузере по умолчанию.

«В QuiteRSS множество функций, но они не лезут вам на глаза.»



► Какое бы действие вам ни понадобилось, в *QuiteRSS* для него найдется кнопка на панели инструментов.

QuiteRSS очень понятный и простой интерфейс: он поддерживает классический и газетный режим просмотра, предлагает расширенные инструменты поиска (в том числе поиск по ссылкам), несколько способов выделения (например, звездочки, ярлыки и фильтры) и множество кнопок тонкого управления для работы с уведомлениями.

В новой версии *QuiteRSS* исправлен ряд ошибок, кое-что по мелочи улучшено и усовершенствован дизайн. Сайт проекта содержит инструкции по установке *QuiteRSS*, и есть множество прекомпелированных пакетов — как минимум, для всех основных дистрибутивов Linux.

Извлечение аудио с CD

Audex

Версия: 0.79 Сайт: <http://bit.ly/AudexApp>

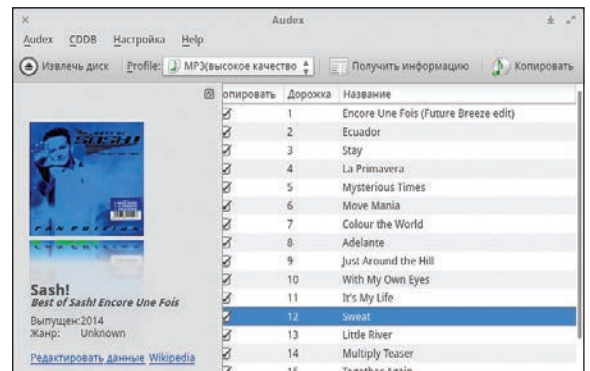
Сдуйте пыль со своих аудио-CD и наконец-то конвертируйте их в музыкальные файлы. Конечно, эта идея была свежей лет десять назад, когда большинство любителей музыки полагались на физические носители. Ныне CD частенько применяются в качестве дотопного способа послушать музыку... или пугать ворон; но все же у большинства из нас до сих пор хранятся десятки дисков. Так почему бы наконец-то не конвертировать их в WAV, OGG, MP3 или даже WMA?

Audex — это приложение для переписывания CD на ПК, ведущее свой род от KDE. Однако оно превратилось в независимое от рабочего стола приложение, и сегодня поддерживает крутые уведомления не только для рабочего стола Plasma, но и для боковой панели Unity и даже для сторонних доков вроде Plank.

Прочитав CD в вашем приводе, *Audex* покажет вам список треков, автоматически осуществит поиск по базе данных

CDDB и применит названия к трекам. *Audex* даже позволит вам решить распространенные проблемы, разделяя названия, меняя местами название и исполнителя, записывая названия с заглавной буквы и автоматически указывая исполнителя.

На левой панели есть специальное поле для отображения обложки. Оно отлично работает, но обложку придется скачивать вручную: автоматически это не делается. Кроме того, придется вручную проверять профили кодировки. По умолчанию *Audex* предлагает только профили WAV, но легко добавить и другие. Перейдите в Settings > Configure Audex... > Profiles [Параметры > Настройка Audex... > Профили] и нажмите на кнопку Search for codecs [Поиск



► Перед списыванием вашей коллекции CD обратитесь к *Audex* для идеального точного контроля за метаданными каждого трека.

кодексов]. *Audex* проверит вашу систему, автоматически определит наличные кодеки и немедленно создаст соответствующие профили. В каждом профиле — множество настроек, где можно указать кодек и скорость передачи и выбрать, хотите ли вы встроить в файл обложку; после чего для переписывания CD достаточно будет нажать на кнопку Copy.

Мы сочли *Audex* отлично поддающимся индивидуальной настройке, и при этом очень понятным и отлаженным. В нем есть весьма продвинутое функции для именования файлов (такие как замена пробелов на нижнее подчеркивание) и местоположений. Например, он может загружать файлы в удаленное местоположение.

«Мы сочли Audex отлично поддающимся настройке.»

Украшения рабочего стола

XDecorations

Версия: 0.1.5 Сайт: <http://bit.ly/XDecorations>

Возможно, ИТ-администраторы старшего поколения помнят о такой штуке, как Microsoft Plus!, которая несколько скрашивала жуткие угловатые GUI 1990-х. Прошло много лет, и теперь у нас есть чудесная программа *XDecorations* для Linux: она предназначена для внесения атмосферы праздника в серые будни вашего рабочего стола.

Мы не нашли прекомпилированных пакетов *XDecorations*, и, похоже, программа распространяется исключительно в виде tar-архива исходника. Но компиляция очень проста. Загляните в файл **Install** в корне директории исходника и, при необходимости, установите требуемые зависимости, например, *libxpm-dev*, *libxt-dev*, *libxext-dev* и *libimlib2-dev*. Затем наберите **make** — и вы уже готовы творить стильные шедевры. Чтобы начать работу с *XDecorations*, воспользуйтесь файлом по умолчанию **xdecorations.rc** в директории проекта. Давайте начнем с готовой темы Xmas [Рождество]:

```
./xdecorations -configfile ./xdecorations.rc
-holiday Xmas -showtinsel
```

Помимо Xmas, с *XDecorations* поставятся еще темы Autumn [Осень], Halloween [Хэллоуин] и Weather [Погода]. В каждой имеется собственный набор декоративных элементов, который отлично редактируется (загляните в папку **Pixmap**). Чтобы увидеть результаты вашей творческой деятельности, просто минимизируйте все окна и наслаждайтесь забавным снеговиком, мерцающими огоньками и кружащимися снежинками. Раз в несколько секунд появляется Санта в санях, запряженных оленями.

Чтобы сменить праздник, откройте файл **xdecorations.rc** и найдите там переменную тем. Измените ее на свой вкус,

«Для внесения атмосферы праздника в будни вашего рабочего стола.»



» *XDecorations* предлагает украшения рабочего стола для разных времен года и праздников.

например, на Autumn или Weather. Файл настройки содержит массу опций, причем у большинства из них имеются примечания в закомментированных строках.

XDecorations совместимы с большинством менеджеров окон, линия лампочек соответствует вашей верхней панели (если она есть), и все вместе не особо сильно нагружает ресурсы — по крайней мере, мы не заметили особого влияния на производительность системы. Новая версия *XDecorations* предлагает улучшенную анимацию, сложное освещение и более натуралистичные эффекты ветра.

Создатель пакетов Debian

Debreate

Версия: 0.7.10 Сайт: <http://debreate.sourceforge.net>

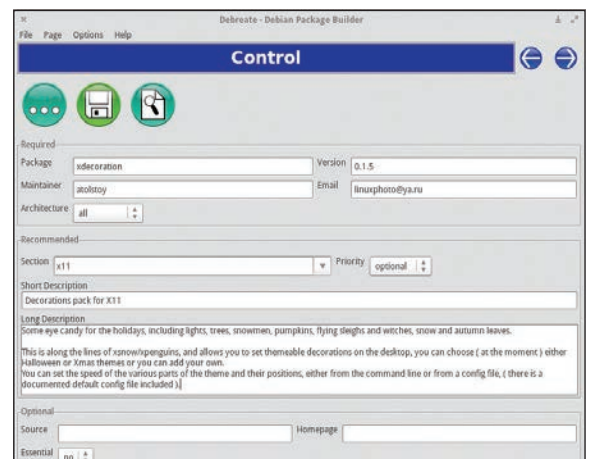
Сборка пакетов для Debian и его производных, например, Ubuntu, Linux Mint и ElementaryOS — занятие привлекательное, благодаря множеству удобных инструментов поддержки. Один из них — *Debreate*, предлагающий графический интерфейс для всего процесса создания Deb-пакета. *Debreate* вдохновлялся приложением Java — создателем пакетов *Packin*. Но *Debreate* удобнее и очень хорошо документирован.

Установка *Debreate* проста. На сайте проекта есть множество файлов, из которых вам понадобится опробованный нами универсальный **debreate_0.7.10_all.deb**, он подходит для всех систем. При первом запуске программы та уведомит вас о местоположении своих настроек (~/.config/debreate) и покажет первый экран приветствия, на котором также помещена ссылка на видеоруководство.

Остальная часть процесса включает несколько шагов в мастере настройки.

Для начала вы заполняете метаданные пакета на экране Control. Здесь есть обязательные для заполнения поля (Package, Maintainer, Version и E-mail), рекомендуемые и опциональные. Далее появляется экран Dependencies, где вы указываете все необходимые зависимости. Категорий здесь немало — можно выбрать рекомендации, конфликты, расширения и некоторые другие. Далее мы выбираем файлы, которые будут включены в наш пакет. Можно изменить префикс установки для определенного файла или применить настройку к папке. Затем мы перейдем на экран Scripts. Здесь доступны сценарии предустановки, пост-установки, предудаления и пост-удаления, что позволяет

«Графический интерфейс для процесса создания Deb-пакета.»



» *Debreate* предлагает простой способ создания Deb-пакета для тестирования на Debian и его производных.

получить отдельный набор скриптов для каждого случая. Следующие три дополнительных экрана добавляют Changelog, Copyright и Menu Launcher соответственно, и все их очень просто заполнить. И, наконец, мы попадаем на экран Build и нажимаем на большую зеленую кнопку, чтобы создать наш Deb-пакет. Если все было настроено правильно, то вы будете готовы к тестированию.

HotGames Развлекательные приложения

Стрелялка от первого лица

Chocolate Doom

Версия: 2.1.0 Сайт: www.chocolate-doom.org

На самом деле, эта игра включает не только *Doom*, но также *Heretic*, *Hexen* и *Strife* — все легендарные стрелялки 1990-х от первого лица. Однако порт *Doom* — главный герой проекта, и мы рассмотрим его поближе.

Мы уже видели ряд портов после того, как iD Software открыла исходники *Doom* в 1998 г., но проект *Chocolate Doom* является попыткой его сохранения и нацелен на поддержание минималистского исходного порта *Doom*, который позволяет игрокам впоследствии использовать разные любительские модификации и дополнения *Doom*. Не секрет, что за все эти годы в архиве idgames скопились тысячи модификаций *Doom* от его фанатов, и многие из них были созданы еще до открытия исходника *Doom*. Каждый раз, когда расширение движка *Doom* с открытым кодом нарушает совместимость

с этими историческими разновидностями, проект *Chocolate Doom* тщательно восстанавливает ее.

Chocolate Doom является в чистом виде игрой GNU GPLv2 с открытым кодом. Однако ей нужны игровые данные, а значит, нужны файлы *Doom WAD*. Она имеется в пакетах многих дистрибутивов Linux, таких, как Ubuntu/Mint, openSUSE, ROSA и прочих. И всякий раз вы увидите пакет *doom-wad-shareware*.

Игра поддерживает все разновидности *Doom*, включая *The Ultimate Doom*, *Doom 2* и *Final Doom*, а также *Chex Quest* и *HACX*. *Heretic*, *Hexen* и *Strife* также

«Выглядит как надо, словно и не прошел уже 21 год.»



► Пospешим в мир низких потолков, где на вас отовсюду нападают агрессивные сгустки пикселей.

поддерживаются *Chocolate Heretic*, *Chocolate Hexen* и так далее. Если у вас есть несколько игр (IWADS), вы можете переключаться между ними с помощью `chocolate-doom -iwad /путь/к/вашему/doom.wad`

Игра начинается в полноэкранном режиме и выглядит как надо, словно и не прошел уже 21 год.

Аркада

VoltAir

Версия: 1.0.5 Сайт: <http://bit.ly/VoltAir>

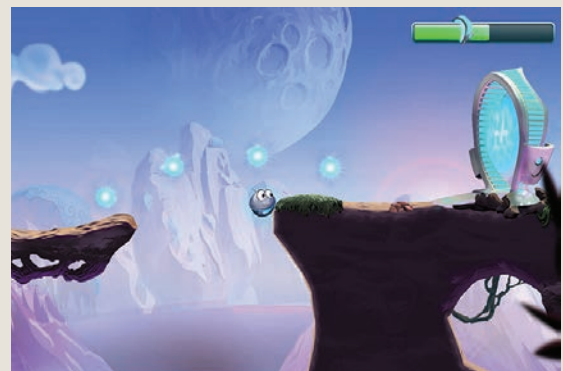
VoltAir — очень интересная и красочная игра; в ней вы на неизвестной планете управляете электро-роботом в форме шара. Во время игры ваш робот собирает заряженные шары, наращивая силу, что помогает ему сбежать через портал в конце каждого уровня. Пользователи могут играть как в одиночку, так и в локальном многопользовательском режиме. Игре присущи профессиональная графика и очень милая фоновая музыка.

Это отнюдь не еще одна игра с открытым кодом на SDL: *VoltAir* на самом деле разработана Fun Propulsion Labs (часть Google), под Apache Public License. Это «кросс-платформенный платформер», написанный на C++ и применяющий для интерфейса библиотеки *QtQuick/QML*. Среди целевых платформ — Linux, Mac OS X и Android, хотя вполне очевидно, что лучше всего игра идет на мобильных

устройствах. Несмотря на свою простоту, игра очень занимательная, однако заставить ее работать на Linux — задача не из легких.

Игра имеется на Google Play (<http://bit.ly/1xo6lmu>), и вы можете запустить ее на рабочем столе Linux с помощью *ARChon* (примочка для *Chromium* из того же источника, см. <http://bit.ly/ARChon>). Чтобы скомпилировать *VoltAir* из исходника, надо ознакомиться с официальным руководством, и сначала скомпилировать *LiquidFun* — C++-библиотеку 2D-физики на основе *Vox2D*, которая включает высокопроизводительную имитацию частиц. Потом вы сможете скомпилировать

«VoltAir: игра с высокопроизводительной имитацией частиц.»



► VoltAir: на диких, но прекрасных чужих планетах вас встретит переменная гравитация.

саму игру в *QtCreator*. Весь набор *Qt* должен быть как минимум версии 5.3.

Прекомпилированных пакетов *VoltAir* для дистрибутивов Linux пока не существует, кроме как для ROSA R4 и OpenMandriva Cooker, и если вы не хотите сами создавать пакет для *VoltAir*, можете использовать эти дистрибутивы, чтобы без промедления насладиться игрой. *VoltAir* довольно быстрая и отзывчивая, и использует исключительно ввод с клавиатуры.

Среда рабочего стола

Budgie Desktop

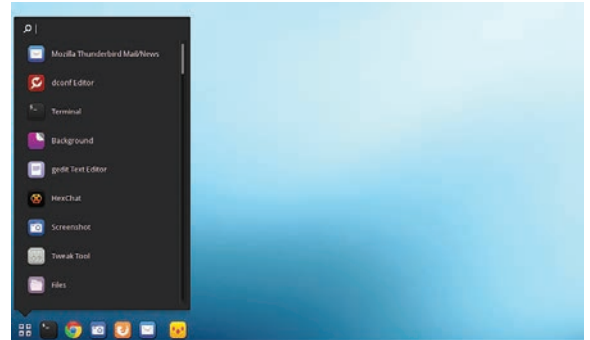
Версия: 0.8 Сайт: <http://bit.ly/BudgieDesktop>

Утверждение, что слишком большого выбора рабочих столов не бывает, элегантно и изящно подтверждает *Budgie Desktop*. Если вы устали от раздутого KDE; вас тошнит от GNOME Shell; вы страдаете от медленной разработки *Xfce* или ненавидите Unity за практическое отсутствие индивидуальной настройки, то вам стоит взглянуть на *Budgie*. Это флагманский рабочий стол EvolveOS, которая, в свою очередь, была скомпилирована с нуля с помощью ветки менеджера пакетов *PiSi*. *Budgie* доступен в готовых к установке пакетах для Ubuntu, openSUSE, Fedora и Arch Linux. Давайте рассмотрим его подробнее.

Разработчики *Budgie* утверждают, что это не ветка GNOME Shell (популярное мнение), а совершенно независимая программа, работающая поверх технологии GNOME и тесно с ней интегрированная. Код написан на C в Vala, с упором на простоту и минимализм. На самом деле, концепция похожа на то, что предлагают команды Yorba

и Elementary OS, однако взаимодействие с пользователем несколько иное.

Основным компонентом после набора разнообразных обоев по умолчанию является нижняя панель, которая похожа на панель ChromeOS. На панели *Budgie* размещаются основное меню, переключатель задач, список окон, системный лоток, регулятор громкости, часы и ряд опциональных апплетов. Каждый апплет можно переместить или удалить и, в большинстве случаев, изменить в диалоговом окне Preferences. Украшениями окон занимается *Budgie Window Manager (BWM)*, ответвление *Mutter* с более ровным управлением анимацией и обоями рабочего стола. Многие другие компоненты, такие, как менеджер сессий или уведомления, сейчас



➤ **Главные украшения Budgie — основные компоненты рабочего стола и живой фон.**

находятся на ранней стадии разработки и служат интерфейсами к соответствующим отлаженным частям GNOME.

В новой версии *Budgie* — десятки улучшений, предлагаемых через 78 недавних поправок. В меню теперь по умолчанию используется более узкая компактная раскладка, и можно настроить его на отображение категорий или просто структуры. Нижняя панель теперь может автоматически скрываться, она поддерживает привязку приложений, динамический выбор тем и многое другое. *Budgie* можно попробовать в сессии live в EvolveOS или прямо в вашей системе. Он унаследовал многие настройки GNOME/Unity/Cinnamon, и желаемым насладиться *Budgie* в чистом виде советуем использовать его в базовой EvolveOS, хотя это только альфа-версия.

«Многие компоненты сейчас на ранней стадии разработки.»

Антивирус

ClamAV

Версия: 0.98.5 Сайт: www.clamav.net

ClamAV — отличное кросс-платформенное решение для определения вредоносного ПО и вирусов. Оно предлагает достаточно высокий уровень обнаружения, и хотя в целом уступает коммерческим антивирусам высокого уровня, но считается удобной и полезной утилитой в мире Linux и BSD.

ClamAV только определяет вирусы и помещает зараженные файлы в папку карантина, но не может удалять их из файлов. *ClamAV* находит вирусы на всех платформах, но особенно он полезен для обнаружения вирусов и вредоносного ПО на Windows, почему и используется для сканирования монтированных дисков NTFS и FAT, разделов SMB и определенных участков, например, папок обмена электронными сообщениями. *ClamAV* доступен практически на всех видах Linux, так что вы можете просто установить его через свой менеджер пакетов. Самый простой и очевидный способ приступить к работе

с *ClamAV* — запустить команду *clamscan*, вот так:

```
clamscan -r /path/to/desired/folder
```

Через некоторое время (зависящее от количества файлов и общего размера папки) *ClamAV* вернет отчет Scan Summary с многословной информацией об известных вирусах, версии движка *ClamAV* и статистике сканирования.

Еще один ключевой исполняемый файл в пакете *ClamAV* — *Bytecode Compiler (clamcc)*, который используется для компиляции сигнатур в байт-код для дополнительной обработки файлов. Сигнатуры *Bytecode* — это специализированный тип сигнатур *ClamAV*, умеющий производить дополнительную обработку

```

x Terminal
/media/atalstoy/7EA2E5C3A2E57FCF/Windows/Resources/Themes/Soft7 2.0/en-US/Soft7
2.0.msstyles.mu1: OK
/media/atalstoy/7EA2E5C3A2E57FCF/Windows/Resources/Themes/Soft7 2.0/Shell/Normal
Color/en-US/shellstyle.dll.mu1: OK
/media/atalstoy/7EA2E5C3A2E57FCF/Windows/Resources/Themes/Soft7 2.0/Shell/Normal
Color/shellstyle.dll: OK
/media/atalstoy/7EA2E5C3A2E57FCF/Windows/Resources/Themes/Soft7 2.0/Soft7 2.0
.opl.msstyles: OK
/media/atalstoy/7EA2E5C3A2E57FCF/Windows/Resources/Themes/Soft7 2.0/Soft7 2.0.n
sstyles: OK
/media/atalstoy/7EA2E5C3A2E57FCF/Windows/Resources/Themes/Soft7 2.0 (top).theme
OK
/media/atalstoy/7EA2E5C3A2E57FCF/Windows/Resources/Themes/Soft7 2.0.theme: OK

----- SCAN SUMMARY -----
Known viruses: 3690212
Engine version: 0.98.1
Scanned directories: 14
Scanned files: 23
Infected files: 0
Data scanned: 4.81 MB
Data read: 4.77 MB (ratio 1.01:1)
Time: 14.912 sec (0 m 14 s)
atalstoy@atalstoy-HP-Pavilion-dv6-Notebook-PC ~ $

```

➤ **Если у вас двойная загрузка с Windows, это неплохой повод использовать сканер *ClamAV*.**

сканированного файла, способствуя более надежному обнаружению.

Свежая версия *ClamAV* улучшила инструменты обнаружения вредоносных программ, включающие в анализ свойства файла и использующие API *ClamAV* и функции *bytecode* для сбора и работы с ними. Кроме того, *ClamAV* теперь поддерживает формат файлов XDP (XML Data Package) для чтения и извлечения, а также декодирования и сканирования файлов PDF внутри файлов XDP. И, наконец, новый *ClamAV* предлагает разнообразные отладки безопасности, предотвращающие сбои в его работе при сканировании определенных вредоносных программ. LXF

«Свежий ClamAV улучшил обнаружение вредоносных программ.»

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 9 ГБ качественного DVD.



Год чего?

Тема года Linux на рабочем столе давно навязла на зубах. Однако какого именно Linux и на каком именно рабочем столе? Есть много-много разных (всяких степеней различия) дистрибутивов Linux, и почти столь же огромное количество рабочих столов, с которыми они могут работать. Одни дистрибутивы являются ответвлениями чего-то, а другие совершенно самостоятельны. Внутри каждого — Linux, но снаружи — полная непохожесть; сколько из них вы уже пробовали?

Инерция во многом определяет наше поведение — именно по этой причине немало пользователей Windows никогда не пытались обратиться к альтернативной ОС; но и пользователи Linux подвержены ее влиянию. Когда вы в последний раз пробовали что-то новое? Я не про загрузку live CD, пару щелчков по значкам и возврат к привычной текущей настройке. Я про попытки попробовать другой рабочий стол. Очень легко установить альтернативу вашему текущему дистрибутиву и выбрать, каким пользоваться, при входе; куда труднее заставить себя придерживаться этого. Возможно, вы уже успели нарушать свои планы и обещания на год, так что попробуйте вместо них другой рабочий стол...

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.

Плоды Enlightenment

ALT Linux

32-БИТНЫЙ

В этом месяце наше Сравнение было посвящено самым популярным средам рабочего стола. Большинство отобранных кандидатов

являются стандартным выбором для целого ряда дистрибутивов, и включены в Fedora 21 Remix. Однако *Enlightenment* не столь широко распространен, хотя и очень любим своими ревностными сторонниками.

ALT Linux — это дистрибутив, о котором говорят сравнительно мало, но у него вышел релиз live с *E19* — последней версией *Enlightenment* — в качестве рабочего стола по умолчанию, и мы решили включить его на DVD для желающих попробовать нечто непохожее на других.

Хотя основной причиной включения ALT Linux был все-таки *Enlightenment*, ALT Linux по праву может рассматриваться именно как дистрибутив (он ответвился от Mandrake), а не как демонстрационная площадка рабочего стола *E19*.



Фаворит безумных шляпников

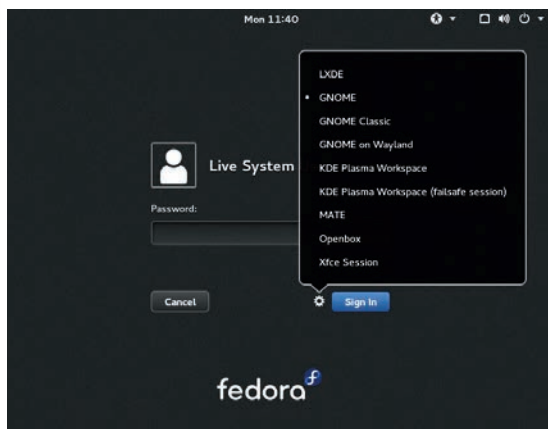
64-БИТНЫЙ

Fedora 21

В этом месяце у нас большая статья о передовых технологиях, которые доступны — или будут доступны — в Linux (статья начинается на стр. 30), а уж если какой дистрибутив соответствует определению «передовой», то это Fedora.

Испытательный полигон Red Hat на базе сообщества включает по умолчанию множество активно обсуждаемых пунктов, в том числе графическую систему *Wayland*. Как ни популярны наши ремиксы Ubuntu с дополнительными рабочими столами, но почему вся любовь должна доставаться им? Итак, мы включили смешанную версию Fedora с рабочими столами KDE, *Xfce*, *Mate* и *LXDE*, добавленными к Gnome, рабочему столу по умолчанию.

Чтобы использовать другой рабочий стол в среде live, выйдите из системы и снова войдите как пользователь системы live с паролем live. Слева от кнопки входа есть небольшой значок, позволяющий выбрать другой рабочий стол, в том числе Gnome на *Wayland*. Щелкните по нему после выбора live user, но до нажатия на Sign In. Этот ремикс Fedora устанавливается обычным способом, и затем можно выбрать предпочтительный для вас рабочий стол по умолчанию, войдя в систему как обычный пользователь.





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.

М-мм, мини

32-БИТНЫЙ

4MLinux

Дистрибутивов сотни; есть большие и именитые; есть респины других дистрибутивов, вызывающие разную степень интереса; и есть ряд независимых дистрибутивов. 4M Linux — любопытный дистрибутив, подпадающий под последнюю

категорию. Он использует менеджер окон *JWM* — это уже заметное отличие — и предлагает приятный и интуитивный интерфейс; применение им *Conky* для предоставления системной информации по рабочему столу — иллюстрация того, как заботливо он продуман.



Нечто иное

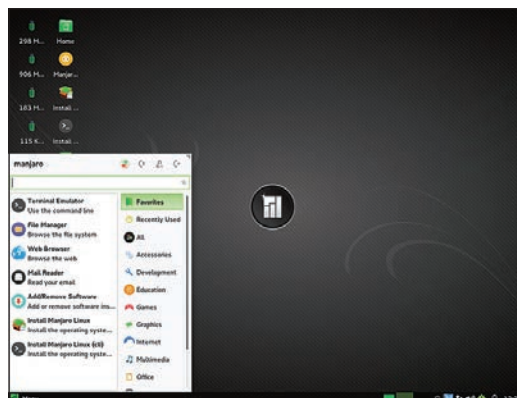
64-БИТНЫЙ

Manjaro 0.8.11

Бывают дистрибутивы явно популярные, но бывают и вездесущие. Все (ну, почти все, кто читает это) слышали о дистрибутиве Debian, но многие работают с его частями, даже не подозревая об этом. Например, Ubuntu основан на Debian, а многие другие дистрибутивы сейчас основаны

на Ubuntu, так что Debian становится всепроникающим, хотя и не трубят об этом.

Arch Linux — еще один дистрибутив, набирающий популярность в качестве основы для создания индивидуально настраиваемых дистрибутивов. Возможности контроля и настройки в Arch Linux делают его идеальной базой для других дистрибутивов, и такие уже созданы. Один из самых популярных — Manjaro Linux, и мы предлагаем вам его последний релиз. Если вам нужна гибкость Arch, но лень тратить время и силы на его изучение — стыдитесь; хотя тогда ваш выход — попробовать Manjaro. Этот дистрибутив доступен в нескольких версиях, различающихся выбором рабочего стола, но «стандартная» его опция по умолчанию — *Xfce*, который мы и предлагаем. Это достаточно популярный выбор, и он стоит внимания благодаря низкому потреблению ресурсов и быстрой работе. Позднее можно установить и другие рабочие столы, весь обычный набор; но сначала познакомьтесь с *Xfce*. **LXF**



И еще!

Бесплатная е-книга

GIMP: Полное руководство

Бесплатно! Мы в *Linux Format* решили пойти чуть дальше, и в этом месяце предлагаем нашим чудесным читателям нечто еще и совершенно даром.

GIMP — это лучшая свободная альтернатива *Adobe Photoshop*, которую вы можете установить на свою любимую версию Linux, и чтобы помочь вам укротить его мощь, мы включили полное руководство по этой популярной программе работы с изображениями на **LXF DVD** этого месяца. Вы познакомитесь с основными навыками, узнаете, как повернуть изображение, обрезать его и настроить его размер, и научитесь добиваться удивительных результатов с помощью работы со слоями, путями и интересными эффектами. В общем, здесь 28 подробных руководств, включающих всё, от создания винтажных снимков, пиксельной графики и монохромных изображений до создания фотомонтажа, реставрации старых снимков и кодирования собственного плагина *GIMP* для создания уникальных эффектов.



Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязанные быть в каждой операционной системе.

Hardinfo Инструмент для тестирования ОС.

Ядро Исходник последней версии ядра.

Memtest86+ Проверьте свое ОЗУ на сбои.

Plop Простой менеджер запуска разных ОС.

RaWrite Создавайте загрузочные диски в Windows.

SBM Независимый от ОС менеджер загрузки с простым интерфейсом.

VvDial Соединяйтесь с Интернетом через телефонный модем.

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:



LXF190
Декабрь 2014

250 руб.

- » Кто на свете всех милее? Дистрибутив-2014
- » Жизнь в Сети Браузер вашей мечты
- » PhotoRec Чьих рук дело — спасение файлов
- » Pi и Mathematica Две знаменитости

LXFDVD: Linux Mint Debian Edition, Trisquel, Mageia, CentOS, Korora, openSUSE, Salix, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_190/
PDF-версия: shop.linuxformat.ru/elxf_190/




LXF191/192
Январь 2015

250 руб.

- » Пиршество идей Проекты для Raspberry Pi
- » Звуки музыки Плейеры для фонотек
- » Робот — друг человека Соберем для дома, для семьи
- » Забыли пароль? Rescatux вас спасет

LXFDVD: все для Raspberry Pi — Raspbian, KaliLinux, PiMusicBox, RetroPie, Jasper, NOOBS; 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_191-192/
PDF-версия: shop.linuxformat.ru/elxf_191-192/



LXF193
Февраль 2015

250 руб.

- » Вещание по дому ПК с Linux как хаб медиа-файлов
- » Векторная графика Объемем контуры на кривых Безье
- » Отстрел диска Не доставайся же ты никому!
- » Горячие точки Визуализация статистики

LXFDVD: OpenELEC 4.95.3, openSUSE 13.2, Ubuntu 14.0, XBMCbuntu 13.2, HandyLinux, Netrunner, Robolinux, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_193/
PDF-версия: shop.linuxformat.ru/elxf_193/

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

ДИСТРИБУТИВЫ

» Содержание

LINUX
ФОРМАТ

4MLinux 10.1 (32-битный)

Абсолютно свободный минималистичный дистрибутив с упором на функциональность.

ALT Linux E16 (32-битный, ISO-образ)

Версия с рабочим столом Enlightenment. ISO-образ не загружается с данного DVD — надо будет прожечь его на CD/DVD либо скопировать на USB-флешку командой dd и загрузить уже с этого носителя.

Fedora 21 (64-битный)

Отлично налаженная рабочая станция — простота в работе операционная система для ноутбуков и настольных ПК, с полным набором инструментов и подручных средств для разработчиков и прочих умельцев.

Manjaro 0.8.11 Xfce (64-битный)

Дружелюбный к пользователю дистрибутив на базе независимо разрабатываемой ОС Arch.

Void 3.0.0 (64-битный)

Дистрибутив на базе Utiliti с элегантным и легковесным оконным менеджером Enlightenment. Поддерживается легковесный приложенный и предлагает высокую степень настраиваемости.

LMDE 2 Sinnamon (64-битный)

Релиз-кандидат популярного дистрибутива, на сей раз на базе Debian.

Oinku April 7.0 (64-битный)

Экспериментальный минималистичский дистрибутив, отвечающий от Parrot Linux.

Tails 1.3 (32- и 64-битный)

Дистрибутив, способствующий анонимности веб-серфинга с помощью сети Tor.

Xubuntu 14.04.04 (64-битный)

Дистрибутив с рабочим столом Xfce.

Окончание на обороте »

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!

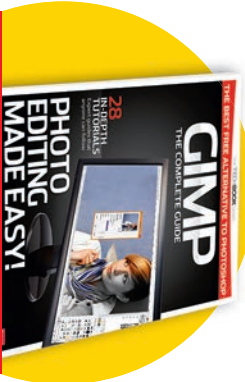
LINUX
ФОРМАТЛУЧШИЕ ДИСТРИБУТИВЫ И БОЛЕЕ ТОГО!
ВСЕ ДЛЯ БЫСТРОГО СТАРТА В LINUX

ФЕДОРА 21

РАБОЧАЯ СТАНЦИЯ

Главные авангардные функции Red Hat

64-битный



GIMP: Редактируем фото за простоту!
БЕСПЛАТНАЯ Е-КНИГА

Manjaro 64-битный
Мощный дистрибутив, посыланный каждому!

Содержание

LINUX FORMAT

БЕСПЛАТНАЯ Е-КНИГА

GIMP Полное руководство (на английском языке)

НОТРИСКИ

- Audex 0.79** Извлечение аудио с CD
- Budgie Desktop 0.8** Среда рабочего стола
- Chocolate Doom 2.1.0** Стрелялка от первого лица
- SlamAV 0.98.5** Антивирус
- Debgate 0.710** Утилита создания пакетов Debian
- Fifth 0.11** Web-браузер
- GNU Hello 2.10** Программа приветствия GNU
- QuiteRSS 0.171** Читалка RSS
- Volt!R! 1.0.5** Аркада
- Xdecorations 01.5** Украшения рабочего стола
- XFDesktop 4.10.3** Менеджер рабочего стола

УЧЕБНИКИ

Файлы для урока по Java Physics
Файлы для урока по SquirrelMail
Файлы для урока по ПЛИС

ПОМОЩЬ

Руководство новичка
Руководства

ГЛАВНОЕ

- CheckKinstall
- Coreutils
- HardInfo
- Kernel
- Memtest86+
- Plop
- SBM
- WvDial

ДОКУМЕНТАЦИЯ — 10 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

- Bash Scripting** Подробное руководство по программированию на Bash
- Bourne Shell Scripting** Начальное руководство по программированию на Bash
- Cathedral Bazaar** Классический текст Эрика Раймонда [Eric S Raymond] «Сбор и базар»
- The Debian Administrator's Handbook** Руководство администратора, написанное разработчиками Debian на Python
- Dive Into Python** Учебник по программированию на Python
- Intro to Linux** Начальное руководство по Linux
- Linux Dictionary** Словарь Linux, объясняющий специфическую терминологию
- Linux Kernel in a Nutshell** Описание ядра Linux, созданное одним из его выдающихся разработчиков — [Грегори 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

Пожалуйста, перед использованием Ассного диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу: disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аналитический сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных нашей программой или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, с нами, чтобы убедиться в его безопасности. Мы благодарим вас за вашу поддержку Linux Format.

Тираж издательства ООО «Марком», 186852, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИДПТР ВАО № 77-03.

Создание установочных дисков при помощи cdcrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdcrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdcrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке — например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdcrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdcrecord*, сохраните некоторые настройки в файле `/etc/default/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

```
cdcrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *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/>

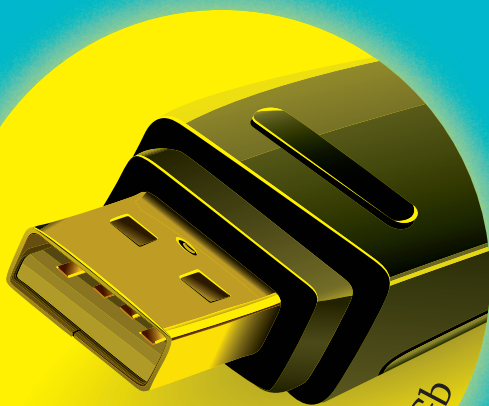
Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

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/

RH124, RH134, RH254
февраль 2015



RHEL7 в формате LVC

Пройди путь от новичка до инженера Red Hat не выходя из дома

**LVC (LIVE VIRTUAL CLASS) -
ИНТЕРАКТИВНОЕ ДИСТАНЦИОННОЕ ОБУЧЕНИЕ
В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ**

RH124 - RED HAT SYSTEM ADMINISTRATION I	02.02.15 - 06.02.15
RH134 - RED HAT SYSTEM ADMINISTRATION II	09.02.15 - 12.02.15
RH254 - RED HAT SYSTEM ADMINISTRATION III	16.02.15 - 19.02.15

WWW.UNIXEDU.RU

ГНУ/Линуксцентр
приглашает на работу!



ВАКАНСИЯ: Разработчик систем электронного документооборота

ТРЕБОВАНИЯ:

- » Опыт внедрения и сопровождения систем электронного документооборота на базе Alfresco
- » Умение описать бизнес-процесс
- » Знание SQL, понимание принципов построения и функционирования баз данных
- » Знание СЭД, отличных от Alfresco, приветствуется

ОБЯЗАННОСТИ:

- » Разработка систем с нуля, сопровождение
- » Настройка системы в соответствии с требованиями бизнеса
- » Написание инструкций для конечных пользователей

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/



В апрельском номере Рванем в Linux!

Со стапелей сошли свежие версии Ubuntu, Mageia и Fedora — ну как такие не установить...

Ящик с Tor

Исследуем, как открытый код и немного электроники помогают сохранить анонимность в Сети.

Поскриптим

Какой скриптовый язык лучше? Разрешим их спор под нашим чутким руководством.

Python 3

Как приступить, портировать свои наработки и перейти на этот новый Python... но пока не на Pi.

Содержание будущих выпусков может меняться — но на прошлый-то выпуск это ведь не влияет?

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 2000 экз., распространение электронной версии 30000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Максим Алибаев, Елена Ессяк, Даниил Кривошеин,

Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова

Редактор диска

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

ООО «Линукс Формат»

Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный,

территория предприятия «Балтика», д. 6/н, лит. Ф

Тел. (812) 462-8383, e-mail: office@ldprint.ru

Заказ 12300

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@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] efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Рассел Барнс [Russell Barnes], Орестис Бастунис [Orestis Bastounis],

Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick],

Крис Браун [Chris Brown], Лайам Браммит [Liam Brummit], Шон Конвей

[Sean Conway], Иэн Дрансфилд [Ian Dransfield], Кент Ельчук [Kent Elchuk],

Мэтт Хансон [Matt Hanson], Дэйв Джеймс [Dave James], Кевин Ли [Kevin Lee],

Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Шашанк Шарма

[Shashank Sharma], Александр Толстой [Alexander Tolstoy], Михалис Цукалос

[Mihalis Tsoukalos], Евгений Балдин, Андрей Гондаренков, Татьяна Иванова,

Михаил Остапкевич, Алексей Федорчук

Поддержка выпуска Алекс Саммерсби [Alex Summersby]

Художественный ассистент Элвин Уйтмен [Alvin Wheelman]

Иллюстрации Шейн Коллиндж [Shane Collinge]

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

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания Linux Format, являются собственностью или лицензируются Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

"GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт www.futureplc.com.



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+

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.

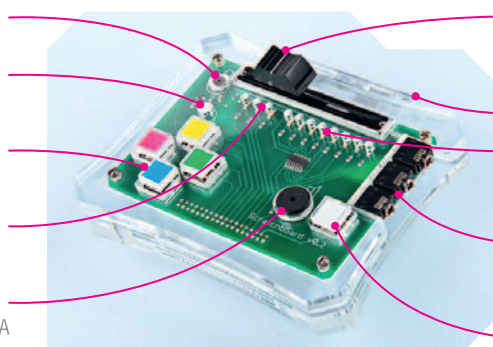
ДАТЧИК ЗВУКА*

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

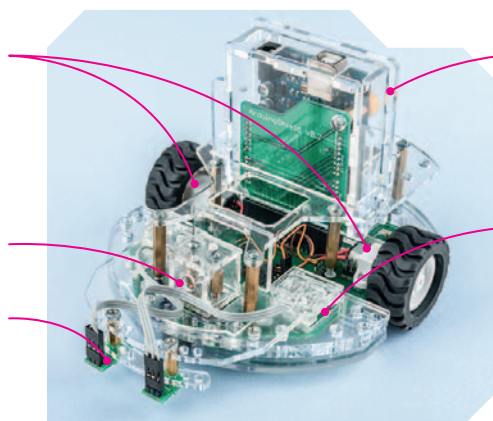
обеспечивает возможность:

- 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)

HETZNER DEDICATED SERVERS

HETZNER
— ONLINE —

В ЦЕЛОСТИ И СОХРАННОСТИ!

ХРАНЕНИЕ ДАННЫХ В ГЕРМАНИИ



ВЫДЕЛЕННЫЙ СЕРВЕР PX60

- Intel®Xeon® E3-1270 v3 Quadcore Haswell
- 32 ГБ ECC RAM
- 2 x 2 ТБ 6 Гбит/с SATA 7200 об/м Enterprise класс Software-RAID 1
- 30 ТБ Трафик*
- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта

2600

рублей в месяц
+ Установка 3700 рублей



ВЫДЕЛЕННЫЙ СЕРВЕР PX70-SSD

- Intel®Xeon® E3-1270 v3 Quadcore Haswell
- 32 ГБ ECC RAM
- 2 x 480 ГБ 6 Гбит/с SATA SSD Data Center Series
- 50 ТБ Трафик*
- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта

3700

рублей в месяц
+ Установка 3700 рублей



Безопасное защищённое хранение ваших данных в собственных дата-центрах немецкой компании Hetzner Online, соответствующее высоким требованиям к стандартам защиты данных в Германии.

RU.HETZNER.COM



MADE IN
GERMANY



100%
Green Electricity
Energy-efficient
Hardware

GreenIT **2011**
Best Practice Award

Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выбирайте более чистое будущее вместе с Hetzner Online!

* Нет платы за превышение. При превышении 30 ТБ/месяц (PX60), 50 ТБ/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 ТБ.