

НА ВАШЕМ БЕСПЛАТНОМ DVD
LXLE: Воскресим старое железо
+ Gm!l, OS/4 и сверх того!

LINUX FORMAT

Главное в мире Linux

Декабрь 2013 № 12 (177)



Мобильная смесь

Свобода у вас в кармане

Plasma Active » Ubuntu Touch » Tizen
» CyanogenMod » Firefox OS » Sailfish



ПЛЮС!
Из биографии
KDE и Gnome
с. 50

Blender Трассируем перемещения с. 84
LIVES Бежим по многим дорожкам с. 88
LMD Защищаем Linux от угроз с. 92



Юные таланты-технари

» Разделите радость созидания с детками из Young Rewired State с. 42



Гуру Google

«Без открытого ПО и Linux, Google могло и вовсе не быть»

Крис ДиБона проповедует свободу ПО в Google с. 38

Также в номере...

Телефон с Firefox

Подлинно открытая ОС на базе Linux для телефона с. 16



Проект Sputnik

Как Dell строит на Linux ноутбук для разработчиков с. 46



Сравнение: читалки новостей

Дуэль похлеще телевизионных с. 24



Пообщаемся

IRC 101

» Болтайте, как мы это делали еще до Твиттера

Предотвратим беду

Open Media Vault

» Сохраните ценные данные, не вникая в BSD

Ощутим всю мощь

Сборка ядра

» Настройте ради скорости, управляемости и потехи

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



Red Hat Enterprise Linux

предоставляет вам **производительность, масштабируемость, безопасность и надежность**, ранее доступные только на очень дорогих платформах

Самая популярная в мире Linux платформа для бизнеса

Обеспечивает высокую производительность, надежность, масштабируемость и безопасность

Сертифицирована ведущими производителями оборудования и разработчиками ПО



Совместима с широким спектром оборудования от рабочих станций до серверов и мэйнфреймов

Обеспечивает одинаковые условия работы приложений при использовании в физической, виртуальной и облачной средах

Пользователи RHEL экономят на оборудовании, лицензиях на программное обеспечение и эксплуатационных расходах



ГНУ/Линуксцентр — Linux-эксперт для вашего бизнеса

- Premier Business Partner компании Red Hat
- 12 специалистов по разработке и внедрению, сертифицированных компанией Red Hat
- Более 100 клиентов, использующих Red Hat
- 10 лет на рынке

Red Hat — ведущий серверный дистрибутив Linux

- Более 15 лет промышленного использования
- Свыше 80% рынка корпоративного Linux по данным CIO Insight
- 5 лет среди лучших вендоров
- Выгодная совокупная стоимость владения (TCO)
- Поддержка в течение 10 лет

Специальное предложение!

Закажите Red Hat Enterprise Linux в ГНУ/Линуксцентре и получите в подарок книгу «Полное руководство пользователя Red Hat Enterprise Linux»



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

15–16 ОКТЯБРЯ САНКТ-ПЕТЕРБУРГ



IX ПРАКТИЧЕСКАЯ
КОНФЕРЕНЦИЯ
ПО HR-БРЕНДИНГУ

РАЗБИВАЕМ СТЕРЕОТИПЫ

Осенняя конференция HeadHunter традиционно имеет практический, интерактивный, острый формат. В этот раз мы хотим обсудить ряд стереотипов, которые мешают нам всем принимать оптимальные решения, быстро реагировать на изменения рынка труда и развивать конкурентные HR-преимущества наших компаний.

HR-брендинговые ошибки и заблуждения

МИФЫ ВОВЛЕЧЕННОСТИ ПЕРСОНАЛА

Важна ли отраслевая HR-специфика?

**Кто «прогнется под изменчивый мир»
поколение Y или X?**

КРОСС-ОТРАСЛЕВЫЕ КЕЙСЫ.

КНУТ, ПРЯНИК, КРП ИЛИ МОТИВАЦИЯ 3.0?

Геймификация бизнес-процессов –
спасение или очередная мода?

**Должны ли мы заниматься
счастьем сотрудников?**

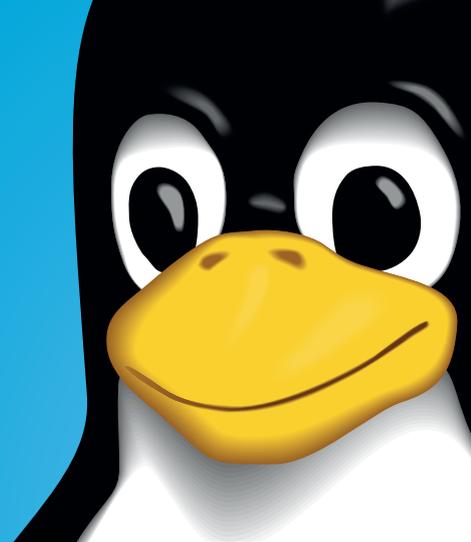
И многое другое...

Стоимость участия по кодовому слову
«Раннее бронирование»
действует только **до 1 августа**.
Обращайтесь к своему персональному
менеджеру НН или по адресу:
hrconf@hh.ru.

Спешите забронировать места,
звоните своему персональному
менеджеру НН!

 +7 812 607 77 29





Используйте свободное ПО — сэкономьте годовой бюджет!

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



**С нашей
помощью
вы сможете**

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



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



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

ГНУ/Линуксцентр предлагает:

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

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

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

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

Valve выкатила игровую платформу для Linux, обещав в 2014-м «родные» игры AAA. И мы спросили наших авторов: какая игра больше всего их порадовала бы?



Крис Торнетт

Значит, на моей машине с Ubuntu появится *Half-Life 3*? Кто бы мог подумать!



Эндрю Грегори

Может, Дэвид Брейбен наконец-то озабочится Linux-версией *Elite: Dangerous*...



Эфраин Эрнандес-Мендоса

Как насчет игры с защитой крепости в Мексике, под названием *Куаутемок*?



Бен Эверард

Tetris, управляемый мозговым импульсом, исходящим от светодиодной матрицы, подсоединенной к моему Raspberry Pi.



Маянк Шарма

Distro Tuxool — игра, где вы громоздите пакет на пакет, создавая новый дистрибутив Linux.



Джонатан Робертс

Я занят — играю в *Sid Meier's Pirates!* На другие игры у меня времени нет!



Майк Сондерс

Я все еще торгую драгметаллами между *Semiess* (около созвездия Стрельца) и *Tivea*. Это вовсе не игра.



Валентин Синицын

Dizzy. Или еще что-нибудь из нетленной классики ZX Spectrum.



Ник Вейч

Отказываюсь покупать любую игру, пока не пройду *Manic Mansion*. Ну вот что надо сделать с этим графином?!



Джюльетта Кемп

Классика 1980-х, *Hacker 2*, требует продолжения, в котором мы взломаем АНБ и тем спасем нашу коллективную душу.



Шашанк Шарма

Наигравшись в *Prison Architect*, я решил сменить карьеру и стать политиком.



Нейл Боткин

Было бы здорово иметь порт *Grand Prix Legends*, а не просто версии, запускаемые в *Wine* и *WineX*.



«Слово и дело»

» Когда компьютеры только появились, они могли выполнять только одну работу — вычислять. И тем полностью соответствовали своему названию. Потом, с изобретением магнитных лент и дисков, к этому добавилась еще одна функция: хранение данных, структурированных в виде базы или неструктурированных — файлов. Чуть позже возникла идея удаленного доступа и объединения компьютеров в сеть, но единственным устройством, с которого мог работать пользователь, оставался «глупый» терминал.

Новая эпоха наступила с появлением «персональных компьютеров», которые по замыслу инженеров компании IBM были всего лишь чуть более умными терминалами. Простейшие задачи подготовки данных и работы с текстами можно было решать, не подключаясь к мейнфрейму. О серьезных вычислениях речь не шла.

С ростом популярности Интернет оказалось, что ПК является идеальным абонентским устройством для этой системы связи. Так у компьютера появилось еще одно назначение. Именно доступ в Сеть, в которой хранятся и обрабатываются данные, стал сегодня основной функцией мобильных устройств — смартфонов, планшетов, очков от Google и других.

А слово «компьютер» из их названий как-то само собой исчезло...

Кирилл Степанов

Главный редактор

» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

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

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

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

Содержание

По сути, у нас у каждого в кармане ЭВМ — спасибо закону Мура.

Обзоры

ZTE Open/Firefox OS 16

За HTML 5 и CSS — будущее, как показывает этот смартфон на Firefox OS.



► Firefox OS — подлинно открытая платформа для смартфонов.

Nvidia Shield 17

Удовольствие от авангарда игр на Android в ваших руках — только бы игр было побольше...



► Играйте в игры на Android в поездах, самолетах и автомобилях.

Feng Office 18

Данный комплект намерен взбудоражить ваши документы, в сотрудничестве. Вз-зз!

Fotoxx 13.09 19

Редактировать изображения с помощью открытого ПО, не признавая в своей слабости к GIMP? Мысль неплохая.

Fuze 20

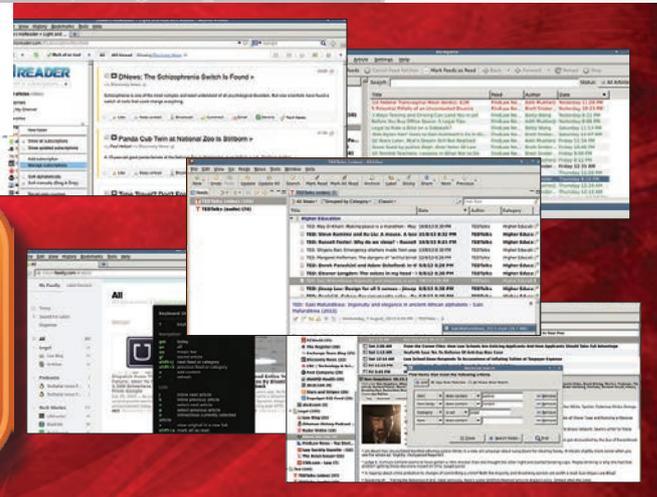
Эта помесь Raspberry Pi и BBC Micro должна бы быть замечательной штукой. А так ли это?

Мобильная смесь

Будущее свободного ПО — у вас в кармане **с. 30**



Сравнение: Читалки новостей **с. 24**



Люди говорят

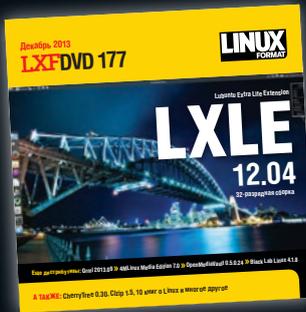


«Думаю, три самых главных достижения — это Android, Chromium и Go»



Крис ДиБона размышляет о достижениях Google **с. 38**

На вашем бесплатном DVD



LXLE

» Ubuntu без настольной шелухи

OS/4 OpenLinux

» Вдохновленная славой AmigaOS

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



Ищите в этом номере



Поколение: Код **42**

Не сомневайтесь: детишки — это фантастика.

Проект Sputnik **46**

Dell выводит Linux на большую дорогу.

На пороге будущего **50**

История рабочих сред дошла до сегодня.



Пропустили номер?

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



Постоянные рубрики

Новости **6**

Сапопикал угодила в Большие Браты, KDE братается с Meeb на общей платформе, Lightworks идет в Linux, а Samsung встал за свободу.

Новости Android **22**

Новый Android опубликован, но атакован троллями, а смартфоны мы скоро будем собирать сами.

Сравнение **24**

83% от полудюжины читалок новостей, готовые принести вам в зубах лучшее из Интернета, как верный пес.

Интервью LXf **38**

Крис Дибона руководствуется только одним принципом: соответствие открытым лицензиям.

Что за штука... Rust **54**

Технология, которая прячется за блистательным Chrome.

Рубрика сисадмина **56**

Доктор твердым шагом продолжает свое головоломное путешествие по регулярным выражениям безумного мира Awk.

Ответы **96**

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик — про избавление от неиспользуемых рабочих столов, защиту портов VNC от хулиганов-скриптописцев, решение проблем со скриптами Bash, установку дистрибутивов с DVD, шпионаж за собственными детьми, хранение медиа-файлов.

HotPicks **100**

Отведайте горяченького: лучшие в мире новинки свободного ПО. 4MLinux, CherryTree, ClassicMenu, Clzip, Equake, Jpgind, Lightspark, Lynx, Mr Rescue, Pragma, Savage Wheels.

Диск Linux Format **106**

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... **108**

Коллекция неполна? Если на вашей полке с журналами образовалась ниша размером с LXf, вы знаете, куда обратиться.

Через месяц **112**

Как краудфандинг меняет ландшафт устройств и программных продуктов.

Учебники

Общение

IRC 101 **62**

Через IRC-канал мы вели войнушки еще без Твиттера. Адски полезен для связи с людьми.

Файловые системы

Делимся с Windows **66**

NTFS, FAT32 и другие способы делить файлы между самой лучшей и самой доходной ОС.

Резервные копии

Open Media Vault **68**

Настройте NAS с дружелюбным Debian — сэкономите свои бесценные фото, видео и музыку.

Продвинутым

Ядро Linux **72**

Перестройте мозг своего ПК с Linux и наградите себя звездпадом на погоны умника.

SDK Ubuntu

Проект QML **76**

Не все нам делать ракеты — иногда можно и колыбельную написать. Баю-бай...

Языки программирования

Erlang **80**

Практикум закончен, но многозадачности еще не конец: эта тема неисчерпаема.

3D-видео

Blender **84**

Привяжем фоновые трехмерные объекты к реальной камере.

Видеомонтаж

LIVES **88**

Сочетаем треки и накидываем на них эффекты... главное — не переборщить.

Безопасность

LMD **92**

Наладим оборону серверов против коварных злоумышленников из Сети. Нет — инъекциям!



ГЛАВНОЕ Скандалы Canonical » Открытая аппаратная платформа
» Linux созрел для кинопрофи » Samsung продвигает Tizen » ZeroNights 2013

КУДА КАТИТСЯ CANONICAL

Компания стала троллем?

Но Марк Шаттлворт уже за все извинился.



» Рубрику готовил
АРТЕМ ЗОРИН

Австрийская премия Большого Брата в этом году присуждена Марку Шаттлворту [Mark Shuttleworth]. По решению жюри, включающего известных юристов, ученых, консультантов, журналистов и борцов за права человека, премия ежегодно вручается госучреждениям, компаниям и частным лицам, чья деятельность вызвала наиболее значительные нарушения конфиденциальности и права на частную жизнь.

Премия обусловлена интеграцией в Ubuntu функции обращения к внешним сервисам при вводе поискового запроса в панели Dash, включенной по умолчанию без явного согласия пользователя, что приводит не только к выборке локального контента, но и к обращению ко внешним сервисам, и данные из поисковой строки могут быть отправлены таким ресурсам, как Amazon, Ebay, Facebook, Flickr, Picasa, Youtube, GitHub, Google News и Wikipedia. В Ubuntu 13.10 эти данные обезличены, а вот в Ubuntu 13.04 владельцы web-сервисов могли сопоставить введенные ключевые слова с IP-адресом пользователя.

Вдобавок Canonical потребовала от создателей сайта fixubuntu.com сменить доменное имя и не использовать на сайте символику проекта Ubuntu, т.к. это нарушает зарегистрированные торговые марки Canonical. Ресурс же fixubuntu.com был создан для распространения инструкции по обеспечению приватности в Ubuntu в интересах пользователей Ubuntu. На сайте, например, говорится, как избежать при использовании «умных областей» в Dash утечки персональной информации.

Создатель сайта fixubuntu.com — Мика Ли [Micah Lee], сотрудник Electronic Frontier Foundation (EFF), технический директор Freedom of the Press Foundation, борец за обеспечение приватности пользователей, специалист по криптографии

и мейнтейнер проекта HTTPS Everywhere. Он автор резонансной статьи о серьезных нарушениях конфиденциальности пользователей Ubuntu.

Торговая марка Canonical явно не упоминает доменные имена, но закрепляет права по использованию слова Ubuntu в области телекоммуникаций, связи и онлайн-сервисов. Использование в продуктах, названиях сайтов и частей URL слов, оканчивающихся на сочетание «BUNTU», требует согласования с Canonical, если это связано с продвижением коммерческих интересов третьих лиц. Ли заявляет, что не нарушил требований, так как слово Ubuntu используется номинативно.

Логотип Ubuntu уже убран с fixubuntu.com, и добавлено примечание, что сайт не связан с Canonical, не отражает ее точку зрения и это независимый проект. Но домен остался прежним, и представители EFF пытаются снять конфликт путем переговоров. Планируется также подготовить инструкцию для защиты от подобных притязаний сайтов — пародий на Ubuntu.

«Инцидент возник не по злему умыслу, а по ошибке.»

Инцидент возник не по злему умыслу, а по ошибке. Шаттлворт извинился в своем блоге за действия юристов Canonical и признал, что попал в довольно глупую ситуацию. Он также подтвердил, что сайт fixubuntu.com не нарушает правила использования торговых марок, которые явно разрешают использование имени Ubuntu и символики проекта для создания сатирических и критических ресурсов. Тем не менее, любой проект, использующий логотип Ubuntu, должен получить



» С ростом популярности и известности компании Canonical растет и ответственность последней перед обществом.

формальное разрешение от Canonical, что необходимо для сохранения контроля за торговой маркой. Юристам Canonical следовало не требовать смены имени домена и удаления логотипа, а лишь пояснить, что владелец сайта обязан спросить на это разрешение.

Но Canonical занимается не только скандалами. В настоящее время компания сотрудничает одновременно с несколькими производителями в направлении подготовки серии моделей смартфонов и планшетов с Ubuntu. Первые смартфоны на базе Ubuntu Touch ожидаются в следующем году. В процессе подготовки Ubuntu 14.04 одним из основных векторов развития станет разработка Ubuntu Touch, редакции для планшетов и смартфонов, которая по умолчанию поставляется с графическим стеком на базе Mir.

Что касается настольного варианта Ubuntu, основные усилия будут направлены на повышение стабильности и производительности рабочего стола на основе Unity 7. Также упоминается намерение выпустить сборку Ubuntu для процессоров на базе архитектуры AArch64 (ARM64). Желаемой целью названо возрождение проекта по обеспечению запуска Android-приложений в окружении Ubuntu.

1С:ПРЕДПРИЯТИЕ 8

Новое поколение систем автоматизации управления и учета

- 1С:Управление производственным предприятием
- 1С:Бухгалтерия
- 1С:Зарплата и управление персоналом
- 1С:Управление торговлей
- Отраслевые и специализированные решения

Теперь работает в Linux!

Для запуска «1С:Предприятия» в Linux необходимо наличие WINE@Etersoft, который входит в состав дистрибутива Mandriva Linux 2011 Powerpack



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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



ЛУЧШЕЕ ОТ СОЗДАТЕЛЕЙ KDE

Новое открытое «железо»

Разработчики KDE и Mer представили OpenHardware-платформу Improv.

Аарон Сейго [Aaron Seigo], основной разработчик KDE-оболочки Plasma, представил новый открытый аппаратный проект Improv, в рамках которого разработана модульная плата для разработки прототипов новых устройств. Платформа развивается сообществом Make Play Live, основанным представителями проектов KDE и Mer, и также занимающимся разработкой планшета Vivaldi. Поступление устройства в продажу ожидается в январе, по цене \$75. Наработки проекта распространяются в соответствии с принципами Open Hardware.

Платформа Improv состоит из двух стыкуемых друг с другом плат — процессорного модуля и функционального модуля. Первый основан на SoC Allwinner A20 с двухядерным ARM Cortex-A7 (1 ГГц) и OpenGLES-совместимом GPU Mali 400MP2, содержит 1 ГБ ОЗУ, 4 ГБ Flash, карту SD, Micro HDMI (видео + аудио),

MicroUSB OTG, USB2 host, шину с i2c, 8 GPIO, UART0. Второй позволяет подключать дополнительные порты и разъемы: например, через замену функционального модуля предоставляется возможность задействования полноразмерных портов USB2, Gigabit Ethernet, SATA, i2c, VGA/TTL и 8 GPIO. Размер платы 85,6×54 мм.

В будущем возможно появление процессорных модулей на базе иных SoC, что позволит разработчику сопрягать имеющиеся процессорные и функциональные модули в зависимости от конкретных текущих задач, изменяя конфигурацию для разных проектов. Спецификации и схемы функционального модуля поставляются под лицензией GPL, что позволяет создавать собственные реализации блоков для стыковки с процессорным модулем. Процессорный модуль разработан по открытому стандарту EOMA68, т.е. совместим с другими устройствами EOMA68.

Improv комплектуется прошивкой на основе Linux-окружения Mer, независимого ответвления от платформы MeeGo. Предусмотрена также возможность установки Debian, Linaro и Android. Окружение Mer составляют системный менеджер *systemd*, пакетный менеджер *zypper*, ядро Linux 3.4 и графические стеки на базе *X.org* или *Wayland*. Для разработки графических приложений предлагаются *Qt4* с *QtQuick1*, *Qt5* с *QtQuick2* и *KDE Platform Libraries 4*.

В качестве графических окружений возможно использование как полноценного настольного варианта KDE Plasma Desktop, так и оболочки для устройств с сенсорными экранами KDE Plasma Active.

Популярность мини-компьютера Raspberry Pi, продажи которого уже превысили два миллиона, свидетельствует о востребованности рынка устройств для энтузиастов-«самодельных». Так что будущее у новой платы явно есть.

ВИДЕО НА LINUX

Легенда теперь и в Linux

Началось бета-тестирование *Lightworks* 11.5 для Linux.

Долгое время Linux недоставало профессиональных программных пакетов для работы с видео и фотографиями. Но ситуация меняется в лучшую сторону. Компания EditShare выпустила вторую публично-доступную бета-версию видеоредактора *Lightworks* для платформы Linux, на кодовой базе *Lightworks 11.5*. Тестирование Linux-версии началось одновременно с Windows-версией. В начале следующего года ожидается релиз *Lightworks 12*, доступный для Windows, Linux и OS X.

Установить *Lightworks* для Linux можно всем, но надо зарегистрироваться на сайте проекта и сгенерировать лицензионный ключ, позволяющий тестировать полнофункциональную Pro-версию продукта 7 дней. После чего официально допускается генерация нового ключа еще на 7 дней. Сборки доступны лишь для 64-разрядных систем и требуют проприетарных драйверов AMD или Nvidia. Официально поддерживаются дистрибутивы Ubuntu 13.10, Linux Mint 15, Fedora 19 и openSUSE 12.3.

Lightworks — профессиональный инструмент, активно используемый в киноиндустрии; он конкурирует с такими продуктами, как *Apple FinalCut*, *Ulead MediaStudio* и *Pinnacle Studio*. Применяющие *Lightworks* монтажеры не раз побеждали в технических номинациях премий «Оскар» и «Эмми». У *Lightworks* удобный интерфейс и непревзойденный набор функций, среди

которых — средства синхронизации видео и звука, наложение разнообразных видеоэффектов в режиме реального времени, «родная» поддержка видео с разрешением SD, HD и 2K в форматах DPX и RED, средства для одновременного редактирования данных, снятых на несколько камер; задействуется GPU для ускорения вычислительных задач. Хотя в Linux-версии *Lightworks* пока реализованы не все возможности, прогресс по сравнению с первым бета-выпуском налицо. Добавлена поддержка декодирования звука AC3. Доведены до полноценного вида диалоги, вызываемые щелчком правой кнопки мыши, и элементы интерфейса для управления воспроизведением. Реализованы функция захвата экрана и возможность определения пользователем параметров цветовой схемы интерфейса. Обеспечено масштабирование интерфейса на экранах сверхвысокого разрешения. Добавлена поддержка выполняемых в фоне операций экспорта и рендеринга. Допускается экспорт материалов на Blu-ray, DVD и YouTube.



» *Lightworks* использовался при обработке видео в таких известных фильмах, как «Король говорит!», «Отступники», «Миссия невыполнима», «Криминальное чтиво» и «Храброе сердце».

TIZEN И SAMSUNG

Кто сказал — Android?

Samsung намерена продвигать свободную мобильную платформу Tizen.

Компания Samsung начала эксперименты с альтернативной Android платформой — Tizen.

В начале ноября был представлен релиз свободной мобильной платформы Tizen 2.2.1, развиваемой под покровительством организации Linux Foundation,

ная поддержка работы на 64-разрядных системах.

Кроме полнофункционального выпуска Tizen для мощных смартфонов, планируется подготовить версию Tizen Mobile Lite, ориентированную на использование в телефонах низшего ценового сегмента. Tizen Mobile Lite сможет работать на устройствах с 256 МБ ОЗУ, 512 МБ Flash и экраном с разрешением HVGA или QVGA.

В конце октября японская компания Systema приступила к ограниченному распространению первого планшетного ПК с Tizen на борту. Планшет пока поставляется только на территории Японии в варианте Tizen Build Kit, нацеленном на разработчиков мобильных приложений. Массовое распространение нового планшета будет осуществляться при поддержке оператора мобильной связи NTT DoCoMo.

Планшет укомплектован 10,1-дюймовым экраном, разрешение которого 1920×1200 пикселей (1080p), 4-ядерным CPU Cortex-A9 1.4 ГГц, 2 ГБ ОЗУ, 32 ГБ Flash, двумя камерами (0,3- и 2-мегапиксельной), слотом для microSD-карт, Bluetooth 3.0, Wi-Fi 802.11n (без 3G-модема). Прошивка основана на выпуске Tizen 2.1. В планшете предусмотрены средства разработки, а в комплект поставки включено также руководство для программистов.

«В комплект поставки включено руководство для программистов.»

при участии LiMo Foundation, Intel и все той же Samsung. Новый выпуск примечателен добавлением в интерфейс настройки меню для управления параметрами, связанными с приватностью. Кроме того, обеспечена возможность установки приложений на SD-карту.

Интересно, однако, что, как представители Samsung раскрыли на конференции Tizen Developer Summit, выпуск первого смартфона на базе Tizen отложен до 2014 года, но первое устройство Samsung с Tizen уже продается: на базе Tizen построена прошивка беззеркальной 20,3-МПикс цифровой камеры Samsung NX-300M, поступившей в продажу в прошлом месяце по цене \$839.

Кроме того, объявлены подробности о возможностях будущего выпуска Tizen 3, который ожидается в третьем квартале 2014 года. Наиболее интересным новшеством является переход с X11 на использование нового композитного менеджера, базирующегося на протоколе Wayland. Tizen 3 будет включать новый фреймворк для разработки визуально насыщенных интерфейсов, отличающийся высокой производительностью и движком рендеринга, способным создавать реалистичную анимацию и манипулировать 2D- и 3D-объектами. В состав выпуска войдет Crosswalk, новый набор компонентов для выполнения приложений на основе технологий HTML5.

В Tizen 3 также будут обновлены системные компоненты и инструментарий для разработчиков. Появится поддержка многопользовательского режима, предоставляющего возможности для работы с мобильным устройством разных пользователей с изоляцией их данных и приложений. Будет обеспечена полноцен-

ная поддержка работы на 64-разрядных системах. Кроме полнофункционального выпуска Tizen для мощных смартфонов, планируется подготовить версию Tizen Mobile Lite, ориентированную на использование в телефонах низшего ценового сегмента. Tizen Mobile Lite сможет работать на устройствах с 256 МБ ОЗУ, 512 МБ Flash и экраном с разрешением HVGA или QVGA.

Для более активного продвижения платформы Tizen и привлечения новых производителей к ее использованию создана организация Tizen Association. В состав ассоциации вошло 36 участников, среди которых компании Panasonic, Sharp, ACCESS, eBay, Nokia Business, HI Corporation, Igalia S. L., McAfee, NTT Data MSE и Open Mobile.

» Первое полнофункциональное устройство на основе Tizen — это вовсе не смартфон!



Фото: <http://images.samsung.com>

НАШИ В БОЮ

Microsoft выиграла

Компания «РОСА» поучаствовала в битве Open Source против Microsoft.

Итак, конференция ZeroNights, о необходимости которой так долго говорили хакеры и специалисты-практики по информационной безопасности (ИБ), состоялась. Она оказалась даже более успешной и яркой, чем ожидалось. Мероприятие собрало более 1000 участников, тогда как сначала организаторы готовились к 700. В первый день было около 900 посетителей, во второй — 600, все четыре зала (главный на 350 мест, второй на 250, два зала для воркшопов на 25 и 50 мест) почти постоянно были заполнены.

Основными темами конференции стали: безопасность систем виртуализации, безопасность АСУ ТП, всевозможные векторы прикладного хакинга, в первую очередь, атаки на критичные бизнес-системы, а также новые методы взлома и глубокие технологические исследования информационных систем. В этом году 45 докладчиков из разных стран мира представили 20 презентаций в основной сессии, 5 уникальных воркшопов, а также 10 выступлений в секции FastTrack.

Эксперты и участники конференции в течение двух дней делились новостями об уязвимостях компьютерных систем и всего, что связано с компьютерами. Поскольку цифровые технологии уже давно вышли за рамки настольных компьютеров и электроники используется везде, то внимание как исследователей, так и злоумышленников переходит на устройства и системы, наиболее тесно связанные с жизнью людей и их благополучием.

Конференция ZeroNights проходит уже третий год подряд. В этот раз перед участниками выступили, в частности, Саханд Ганон [Sahand Ghanoun] с докладом «Дорогая, я дома! Взлом систем домашней автоматизации от Z-Wave», Иван Новиков, он же Владимир 'd0znpp' Воронцов, рассказавший о практике атак на файловую систему с подсчетом времени; Андрей Данау осветил тему наиболее частых ошибок в управлении сессиями в облачных решениях и на классических хостингах. Эксперт Дмитрий Бумов рассказал об уязвимости в логике работы web-приложений панелей управления хостинг-провайдеров, а его коллеги по цеху Владимир Кропотков и Виталий Четвертаков — о практике применения методов и теории распознавания образов для обнаружения атак.



➤ Это было интересно и захватывающе; ждем следующей дуэли!

Кульминацией ZeroNights стало ток-шоу «Хакеры vs Microsoft vs Open Source». Андрей Бешков из Microsoft и Владимир Рубанов из РОСА, которого поддерживали и другие члены сообщества Open Source, сразились в словесном поединке и ответили на вопросы хакеров относительно безопасности ОС. По итогам голосования, MS в лице Бешкова выиграл дуэль, поскольку его позиция выглядела более аргументированной.

Тема ток-шоу — хакеры против вендоров, вендоры против вендоров, СПО (Open Source) против проприетарных решений. Можно ли победить в этой игре и как скоро может появиться надпись “Game over”? Ответы на эти вопросы искали в ходе интеллектуальных сражений представители известнейших ИТ-компаний 8 ноября на ZeroNights 2013.

Никто не верил, но оказалось, что в России можно провести бюджетную платную (!) глубоко техническую хакерскую конференцию по информационной безопасности, собрав более 1000 посетителей. Партнерами стали CareerLab и Яндекс.

ZeroNights — международная конференция, посвященная практическим аспектам информационной безопасности, рассказывает о новых методах атак и угрозах, показывает возможности для нападения и защиты, предлагает нестандартные методы решения задач ИБ. ZeroNights собирает экспертов, специалистов-практиков по информационной безопасности, аналитиков и хакеров со всего мира. В этом году конференция прошла 7 и 8 ноября в московском деловом центре «Коворкинг 2.0». LXF

Новости короткой строкой

➤ Представлен релиз дистрибутива OpenMandriva Lx 2013.0. Это первый выпуск, созданный силами сообщества, после того как компания Mandriva S.A. передала управление проектом в руки некоммерческой организации OpenMandriva Association. Источник: <http://blog.openmandriva.org/>

➤ Марк Шаттлворт объявил о решении отложить переход настольной редакции дистрибутива Ubuntu на Mir и Unity 8 до следующего промежуточного выпуска. Источник: www.omgubuntu.co.uk

➤ Ричард Столлмен предложил снабдить текстовый редактор Emacs возможностями полноценного текстового процессора, поддерживающего визуальное редактирование документов в режиме WYSIWYG. Источник: <https://lists.gnu.org/archive>

➤ Компания Valve продемонстрировала первые образцы игровой консоли на базе операционной системы SteamOS. Были показаны лишь внешние концепции: начинка пока находится на стадии разработки и не афишируется. Имеются данные, что консоль построена с использованием процессора Intel i7 и видеокарты NVIDIA GeForce GTX Titan. Источник: www.theverge.com

➤ Проект Neo900 по созданию преемника коммуникатора N900 запустил кампанию по сбору средств на рабочий прототип устройства. Необходимо будет собрать €25 000. Источник: <http://neo900.org/>

➤ Администрация швейцарского округа Берн заявила, что ПО, разработанное или купленное на средства налогоплательщиков, должно распространяться свободно в открытом виде. Источник: <https://joinup.ec.europa.eu>

➤ Компания Motorola представила новую платформу для создания модульных смартфонов — Project Ara, развиваемую в соответствии с принципами Open Hardware. Источник: <http://motorola-blog.blogspot.nl>

➤ Сообщество Mozilla сообщило о начале продаж в Бразилии смартфонов на базе Firefox OS OS 1.1 и скором старте продаж в Мексике, Перу и Уругвае. Кроме уже распространяемого в других странах аппарата Alcatel One Touch Fire, на рынок Латинской Америки выпущен новый смартфон — LG Fireweb. Источник: <https://blog.mozilla.org>



CEE-SECR 2013

Прошла независимая международная конференция «Разработка ПО/CEE-SECR 2013».

24 и 25 октября в Москве прошла 9-я независимая международная конференция «Разработка ПО/CEE-SECR 2013». В работе конференции приняли участие 790 IT-специалистов из 15 стран мира: от программистов и пред-

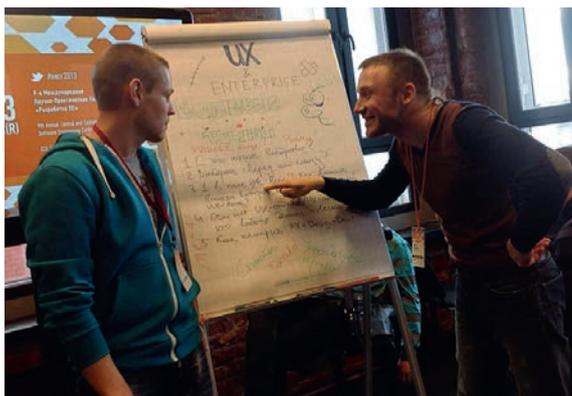
ставителей академического сообщества до предпринимателей и инвесторов.

Программу двух дней составили более 100 докладов и панельных дискуссий. Участники смогли посетить открытое заседание «Руссофт» и предложить тему для обсуждения в новом Conference Discussion Corner. Накануне конференции, 23 октября, прошло 5 мастер-классов.

Местом проведения вновь стал центр новых технологий Digital October. Доклады шли одновременно в пяти залах, заставив участников выбирать наиболее интересную для себя тему. Холлы оживились открытыми дискуссиями, обсуждениями услышанных выступлений и общением с докладчиками.

По отзывам участников, наибольший интерес вызвали:

- » Дискуссия: «Как вывести локальный стартап на глобальный рынок?» — Модератор: Николай Пунтиков, First Line Software.
- » Доклад «LLVM и Clang — новейшие компиляторы и инструменты программиста» — Крис Латтнер [Chris Lattner], LLVM, Apple.
- » Доклад «How To Do Kick-Ass Software Development» — Свен Петерс [Sven Peters], Atlassian.



» Дискуссии были интересными и насыщенными.

О конференции «Разработка ПО/CEE-SECR»

Разработка ПО/CEE-SECR (www.secr.ru) — ежегодная независимая IT-конференция, посвященная проблемам разработки программного обеспечения и программной инженерии, участниками которой становятся разработчики ПО, исследователи, руководители, предприниматели. Впервые состоялась в 2005 году. Конференция SECR завоевала авторитет одной из ведущих IT-конференций, проводимых в России, и традиционно поддерживается мировыми лидерами индустрии высоких технологий.

В состав программного комитета конференции входят эксперты мирового уровня, представляющие как ИТ-индустрию, так и научно-иссле-

довательские организации. Среди них: профессор СПбГУ и генеральный директор «Ланит-Терком» Андрей Терехов, академик РАН Виктор Иванников, генеральный директор Российской венчурной компании Игорь Агамирзян, соучредитель и главный архитектор CUSTIS Максим Цепков, профессор Высшей политехнической школы в Цюрихе Бертран Мейер [Bertrand Meyer] и многие другие.

Основная аудитория конференции — программисты, системные аналитики, инженеры качества, архитекторы программного обеспечения, кадровые специалисты, менеджеры проектов, руководители производства, технологические предприниматели, представители академических кругов, зани-

мающиеся исследованиями в области информатики, а также студенты и аспиранты профильных специальностей.

Программа конференции составляется из отчетов о практической работе, результатов научных исследований, круглых столов, семинаров, приглашенных и ключевых докладов. При отборе докладов программным комитетом используется процедура двойного слепого рецензирования. По результатам конкурса отбирается порядка 35% из поданных на рассмотрение докладов, и их авторы освобождаются от уплаты организационного взноса. С 2005 года организатором конференции выступает ассоциация «Руссофт».

» Доклад «Осторожно: патентные тролли! Теория, практика и актуальные примеры» — Юрий Бардмессер, Bardmessenger Law Group.

» Доклад «WebRTC: От звонков через браузер к новым возможностям вашего бизнеса!» — Владимир Белобородов, МЕРА НН.

По итогам конференции Программный комитет определил победителей Премии Бертрана Мейера [Bertrand Meyer Prize] за лучший исследовательский доклад по программной инженерии. Ими стали Владимир Ицыксон и Андрей Баженов из Санкт-Петербургского государственного политехнического университета с докладом «Прогнозирование характеристик программных проектов с помощью мета-моделирования».

В этом году конференция состоялась благодаря поддержке Российской венчурной компании, Intel, JetBrains, Luxoft, Deutsche Bank, First Line Software, EMC, Oracle, Atlassian, Parallels и IBM. Активно поддержали конференцию также РУССОФТ, АП КИТ, РАЗК, Reksoft и многие другие.

«Мы довольны результатами конференции», прокомментировала Анна Макеева, директор по маркетингу ПингВин Софтвэр.



» Главное в таких мероприятиях — чтобы участники не скучали. И скучать не пришлось.

В целом, мероприятие прошло успешно. Не было пустой болтовни и переливания из пустого в порожнее. Только дело, только конкретика. Презентации, видеозаписи и фотографии постепенно появляются на сайте конференции: www.secr.ru. **LXF**

Об ассоциации «Руссофт»

Некоммерческое партнерство «Руссофт» — результат слияния в мае 2004 года Консорциума «Форт-Росс» с Национальной ассоциацией разработчиков программного обеспечения. «Руссофт» является крупнейшим и наиболее влиятельным объединением компаний-разработчиков программного обеспечения России. Центральный офис Партнерства находится в Санкт-Петербурге. Под управлением Руссофт объединены усилия различных компаний-разработчиков, предлагающих самый широкий спектр технологий и решений в области ИТ.

В «Руссофт» входит более 70 компаний, работающих в области информационных технологий; их штат — более 17 000 высококвалифицированных сотрудников с высшим образованием в области программирования и прикладной математики. «Руссофт» тесно сотрудничает с ведущими университетами, студенты которых неизменно занимают лидирующие позиции в международных олимпиадах по программированию, таких как ACM International Collegiate Programming Contest.

«Руссофт» поддерживает высочайший уровень мировых стандартов по профессиональному обу-

чению, информационной безопасности и контролю за качеством. Компании — члены «Руссофт» являются абсолютными лидерами по внедрению CMMI/ISO в Европе. Среди партнеров и заказчиков «Руссофта» — такие компании, как IBM, Boeing, SAP, Procter & Gamble, CSC и многие другие.

Компании-члены «Руссофт» имеют опыт реализации как малых проектов, так и проектов, измеряющихся сотнями человеко-лет.

НП «Руссофт» — лидер индустрии, ведущий активную борьбу за продвижение интересов компаний-разработчиков ПО перед правительством Российской Федерации и выступающий за развитие ИТ-сектора экономики, снижение налогов, совершенствование таможенного законодательства, построение сети современных технопарков по всей России и привлечение государства к поддержке международных проектов Партнерства.

«Руссофт» ведет маркетинговую и PR-деятельность в России и за границей — в США, Европе и на Дальнем Востоке. Среди наиболее ярких событий — организация коллективных стендов России и стран СНГ на выставках CeBIT (2001–2008),

Comdex Scandinavia (2001–2002), а также roadshows в США (2002–2005), Германии и Скандинавии (2002–2005), ежегодные конференции Software Development Summit (2001–2009), Russian CIO Summit (2003–2008), Mobile Software Forum (2008–2009).

Будучи крупнейшим объединением индустрии, «Руссофт» предоставляет самую точную и достоверную информацию о сфере разработки программного обеспечения в России.

Самостоятельно или совместно с наиболее авторитетными российскими и международными компаниями (например, CNews, IFC) «Руссофт» проводит регулярные маркетинговые исследования, а также поддерживает и постоянно дополняет подробную базу данных по компаниям-разработчикам программного обеспечения в России. Для всего ИТ-сообщества НП «Руссофт» является наиболее надежным источником информации об индустрии.

«Руссофт» входит в состав Ассоциации Информационных и Компьютерных Технологий (АПКИТ), где представляет Комитет по разработке и экспорту программного обеспечения.

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

ССПТ-2 — это сертифицированное ФСТЭК, ФСБ и ГАЗПРОМСЕРТ средство защиты информации нового поколения, реализующее функции межсетевого экрана, но при этом остающееся «невидимым» для любых протоколов и тестовых воздействий, что достигается за счет отсутствия физических и логических адресов на его фильтрующих интерфейсах. ССПТ-2 невозможно обнаружить никакими известными средствами удаленного мониторинга сети.

Скрытность функционирования межсетевого экрана повышает надежность системы защиты в целом и существенно упрощает процедуру установки ССПТ-2 в компьютерные сети и функционирующие на их основе информационные и телематические системы.

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

LINUX FORMAT

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Слово о Cinnamon'e

В историческом цикле, завершившемся в этом номере, я упомянул Cinnamon — форк Gnome 3, созданный в рамках дистрибутива Mint, деривата Ubuntu. Первоначально эта среда представляла собой попытку придать Gnome Shell'у облик классического Gnome 2, заодно снабдив ее средствами настройки, которыми Gnome 3 не блистал. Однако развитие последнего шло в направлении, для разработчиков Mint'a неприемлемом. И им пришлось отказаться от базиса Gnome — в октябре сего года вышла полностью пересобранная среда Cinnamon 2, где большинство базовых компонентов «третьегонома» заменено своими аналогами. В отлаженном виде она войдет в Mint 16, релиз которого, надеюсь, будет доступен ко времени выхода журнала. Однако уже сейчас ее можно опробовать — и в составе Mint 15, и, с рядом оговорок, в Ubuntu 13.10. И опробование это оставляет самое благоприятное впечатление — в частности, от прекрасных средств настройки: богатство, не уступающее конфигуратору KDE, совмещается в них с простотой центра управления старого Gnome 2. И, что характерно, настраивается все без сторонних «твикеров», избытием которых успела прославиться Ubuntu с ее средой Unity. Не проявился в Cinnamon'e и баг с переключением раскладок, недавно столь повеселивший пользователей Saucy Salamander.

Очередной идеал? Нет, просто еще один островок спокойствия и здравомыслия в бурном море тотального «подмикитного» прогресса.
alv@posix.ru

Сегодня мы рассматриваем:

ZTE Open 16
Величие Firefox несомненно. В отдаленном и туманном прошлом он изменил мир Интернета, вернув Сеть миллионам людей — и сейчас пытается сделать то же самое для смартфонов.

Nvidia Shield 17
Перед нами разворачивается новый раунд консольной войны, но на сей раз в нее ввязались несколько необычных противников. Этот контроллер/консоль/стример игр — именно из таких.

Feng Office 18
Вознесется ли этот онлайн-комплект, чтобы, подобно колоссу, стать

на LibreOffice и дотянуться до Google Docs? Или будет позорно заперт, как в Кавдинском ущелье?

Fotoxx 19
Упорядочите свои фото... а также раскрасьте, ретушируйте и ранжируйте. Замечательно и для профессиональных фотографов, и для собирателей снимков котят.

Fuze 20
Взяв Raspberry Pi и корпус в стиле BBC Micro, вы в итоге получите Fuze. Он заставит нас вздохнуть по былым дням квадратных коробок. Запалите Quattro!



► Shield [англ. щит] от Nvidia: от мечей и копий практически бесполезен.

Fuze



► А когда-то мы находили такой дизайн эстетичным... это ж сколько нам уже лет?!

Feng Office

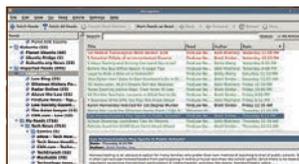


► Выживет ли очередной офисный комплект с новыми приемами работы? Не исключено, что и да.

Сравнение: Читалки новостей

с. 24

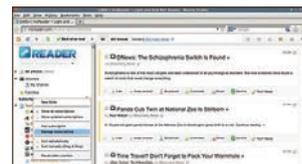
Akregator



Feedly



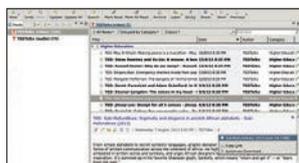
InoReader



Liferea



RSSOwl



Кончина Google Reader оставила в жизни многих людей дыру в форме Really Simple Syndication (RSS, реально простая синдикация). Выберите себе новый клиент, который его создатели не похоронят!

ZTE Open

Это первый на рынке смартфон на Firefox OS, но есть ли чему радоваться? Лес Паундер прикидывает.

Вкратце

» Телефон начального уровня на базе Firefox OS См. также: Android и iPhone.

Спецификация

- » CPU ARM 1 ГГц
- » ОЗУ 256 МБ
- » ПЗУ 512 МБ
- » Карта microSD (до 32 ГБ)
- » Экран 3,5"
- » Камера 3 Мпикс

В мире ныне существуют миллионы смартфонов; соотношение использования в них Android и iOS — почти 50:50, и лишь немногие делают выбор в пользу Windows Phone.

Mozilla недавно выпустила на арену нового претендента: Firefox OS, в основе которого лежит Linux, а приложения написаны на HTML5, JavaScript и CSS. Однако для ОС нужна и аппаратная часть, и так, в сотрудничестве с ZTE (китайским производителем телефонов), появился первый бюджетный смартфон на базе новой Firefox OS.

Трубка

ZTE Open представляет собой моноблок, длина которого составляет 11,4 см, ширина — 6,2 см, а толщина — 1,2 см. Весит он средние 140 граммов, размер экрана — 3,5 дюймов (320×480 точек, это около 165 пикселей на дюйм). Смартфон продается уже разблокированным, и ловит местные сети. Встроенная память — 256 МБ, и хотя в сравнении с 2 ГБ Nexus 4 и ему подобных это выглядит несолидно, для других смартфонов начального уровня это нормально и достаточно для приемлемого качества Firefox OS.

В телефоне использован однокорневой процессор ARM с частотой в 1 ГГц, и 512 МБ системной памяти для ОС. Возможности подключения довольно ограничены, но есть Bluetooth 2.1, Wi-Fi и 3G (HSDPA), которые понадобятся вам, чтобы настроить аппарат в соответствии с инструкциями ваших провайдеров. Имеется встроенная камера в 3 Мп, делающая вполне достойные снимки при условии хорошего освещения, но отсутствие вспышки означает, что фото, сделанные в помещениях, могут быть тусклыми.

ОС еще очень молода; версия, установленная в ZTE Open, до сих пор активно развивается, отсюда большое количество доступных обновлений. Но главное преимущество Firefox OS над всеми прочими мобильными ОС в том, что она открытая.

При первой загрузке вам предлагается настроить телефон для своего региона, что займет всего несколько секунд. На данном этапе, единственный доступный способ импортировать контакты — использовать Facebook или SIM-карту. Если вы переходите на Firefox OS с телефона



» ZTE Open — первый телефон на базе Firefox OS.

Android, как и мы, следует посетить Firefox Marketplace и установить приложение для импорта контактов из Gmail.

Интерфейс Firefox OS приятен и прост. Традиционная кнопка меню заменена пролистыванием: проводя по экрану слева направо, вы переходите к списку установленных приложений, справа налево — к элементам быстрого поиска, таким как отели, карты и т.д. Листая сверху вниз, вы откроете уведомления и быстрый доступ к Wi-Fi, 3G и другим настройкам.

Firefox OS

Firefox OS отличается от других мобильных ОС своими приложениями. Традиционно, приложения для Android создаются при помощи специальной среды разработки (IDE), на языках типа C, Java и JavaScript. Затем эти приложения устанавливаются через специальный магазин приложений.

В Firefox OS приложения написаны с помощью HTML5, CSS и JavaScript, и тем, кто владеет этими популярными языками, легко делать это самим. Еще одна интересная возможность — то, что традиционные web-приложения теперь могут работать автономно через локальное хранилище HTML5, пусть и в ограниченном режиме.

ZTE Open — не самый супер-модный смартфон, но, безусловно, выгодная

альтернатива. На ZTE eBay он продается за £59,99 — весьма щадящая цена по сравнению с другими смартфонами начального уровня, такими как Samsung Galaxy Ace.

Технические возможности у него базовые, но он достаточно функционален для такой стоимости. У нас сложилось впечатление, что новая ОС проста в использовании и не перегружена всякой всячиной. Но количество приложений пока что очень невелико, да и команда Firefox OS еще в стадии формирования. **LXF**

LINUX FORMAT Вердикт

ZTE Open

Разработчик: ZTE / Mozilla
Сайт: www.ztedevices.com
 и www.mozilla.org
Лицензия: Mozilla Public Licence (MPL)

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

» Хороший смартфон начального уровня; стоит того, чтобы купить и разрабатывать приложения для Firefox Marketplace.

Рейтинг 7/10

Nvidia Shield

Дэмиен Мак-Ферран изучает игровое устройство Android, на котором пойдут и игры для ПК.

Вкратце

» По сути, планшет Android, но с классическим игровым интерфейсом. Shield идет ноздря в ноздю с Ouya, Wikipad и GameStick.

Э тот год можно смело назвать годом, когда Android изо всех сил старался покорить геймеров.

Мы уже стали свидетелями революции в сфере микро-консолей после запуска Ouya и в связи с грядущим выходом Catz Mojo и Game Stick, но Nvidia Shield, возможно, станет главным игровым прорывом Android.

Shield — это портативная игровая приставка на базе Android 4.2, гордая своим процессором Tegra 4, сделавшим ее одним из передовых образцов мобильных технологий. Вся эта силища заключена в довольно необычную оболочку, делающую Shield похожей на стандартный контроллер Xbox с воткнутым в него 720-пиксельным, 5-дюймовым сенсорным (емкостным) HD-экраном. Он массивен и тяжел, но удивительно удобен для продолжительного использования.

Классический игровой интерфейс, включающий два аналоговых джойстика, четыре кнопки управления, 8-позиционный D-pad, две боковые кнопки и два аналоговых спусковых крючка, обеспечивает полный арсенал средств управления игрой. Стрелялки от первого лица вроде *Dead Trigger* идут прекрасно, а чувствительный D-pad идеально подходит для ретро-игр, запускаемых в режиме эмулятора.

Мобильные игры становятся все более сложными и требовательными в плане графики, но при таких технических характеристиках очевидно, что Shield справится со всем, что только ни есть в магазине Google Play. Такие игры, как *Crazy Taxi*



и *Real Racing 3*, идут более гладко, чем на любом из виденных нами устройств Android, хотя на основании этих двух примеров можно выявить общую проблему этой консоли: *Crazy Taxi* был обновлен, чтобы поддерживать средства управления Shield, в то время как *Real Racing* — на момент написания — не был. На самом деле, подавляющее большинство скачанных игр не поддерживают систему Nvidia, но будем надеяться, что это исправится, по мере увеличения числа их пользователей.

Воплощая мечты

Между тем, игры Android — только одна сторона Shield. Если внутри вашего компьютера прячется совместимая видеокарта Nvidia, вы также можете транслировать игры с ПК на консоль. Среди поддерживаемых вариантов — *Crysis 3* и *Call of Duty: Black Ops 2*, и сам процесс идет на диво гладко: просто подключаетесь к своему ПК через беспроводное соединение и запускаете игру. Это уникальный коммерческий аргумент, дающий Shield преимущество перед многими другими классическими приставками.

Когда вы не играете, вы можете использовать сенсорный экран Shield для работы со стандартной операционной системой Android. Эта консоль действует как любой другой планшет, позволяя бродить по Интернету, смотреть фильмы HD-качества, читать книги и отправлять письма, хотя форм-фактор, признаться, иногда сказывается, придавая выполнению многих из этих задач известную неуклюжесть.

Ресурс аккумулятора Shield показался нам скорее приемлемым, чем поразительным, составив восемь часов без подзарядки. Также в Shield есть 16 ГБ внутренней памяти и слот для карты microSD, чтобы расширить эту память, а возможность HDMI-подключения превращает приставку в полноценный домашний игровой ПК.

» А вот и не смотрится как контроллер Xbox с воткнутым экраном. Честно.

С технической стороны, придаться в Shield практически не к чему — но ключевая проблема на данном этапе в том, что очень малое число игр используют все предоставляемые возможности управления. Учитывая широкий спектр игр Android, навряд ли многие разработчики станут писать что-то специально под систему Nvidia; скорее он пополнится чем-то наподобие *Asphalt 8: Airborne* от Gameloft, доступными и для других сенсорных устройств, но поддерживающей Shield уже со дня выхода.

Nvidia Shield — великолепная игровая система с невероятными спецификациями и возможностью транслировать игры с ПК, в качестве бонуса. Однако в ближайшие месяцы нам предстоит еще посмотреть, хорошо ли она поддерживается. Пока можно сказать, что у нее удачная конструкция и большая мощь, и, безусловно, она должна хорошо продаваться. **LXF**



Свойства навскидку



Новое слово в играх

Можно транслировать новейшие игры для ПК прямо на 5-дюймовый экран Shield.



Полный контроль

Обилие средств управления игрой в Shield просто несравненное.

LINUX FORMAT Вердикт

Nvidia Shield

Разработчик: Nvidia
Сайт: <http://shield.nvidia.com>
Цена: \$299

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

» Nvidia Shield выводит игры Android на новый уровень — побольше бы только игр.

Рейтинг 8/10

Feng Office CE 2.3.1

Не возлагавший больших надежд на инструмент, названный изначально *OpenGoo*, Маянк Шарма был приятно удивлен.

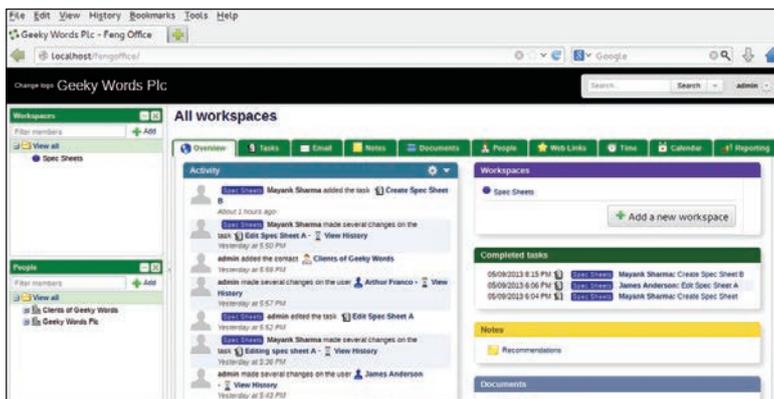
Вкратце

» Система управления проектами. См. также: *TaskJuggler*, *Collabtive*, *ProjectPier*.

Feng Office является частью совместного ПО, разработанного для улучшения эффективности работы группы людей и предоставляет комбинацию функций организации документооборота и управления задачами и взаимоотношениями с клиентами (CRM), плюс несколько совместных элементов: электронная почта, календарь, события, заметки, контакты, документы и задачи.

Построенный с помощью свободных и бесплатных инструментов, Community Editor (см. далее) данного ПО доступен для бесплатной загрузки. Его можно установить на свой web-сервер как любое другое web-приложение LAMP. Инсталлятор на базе web в четыре шага проверяет разрешения на запись в определенных папках, соединяется с базой данных MySQL, потом создает таблицы и завершается, спросив разрешение на создание учетной записи администратора.

Панель инструментов *Feng Office*, называемая Overview, вкратце отображает всю необходимую информацию по системе: незавершенные задачи, последние обсуждения по электронной почте и действия согласно календарю. Работает *Feng Office* по принципу Workplaces [рабочих пространств], т.е. автономный экземпляр для конкретного клиента, проекта или отдела, где по умолчанию установлены модули Почта [Mail], Документы [Documents], Заметки [Notes] и Задачи [Tasks]. Другие функции активируются в настройках администратора. Во вкладке Задачи определяют и назначают задания, их временные рамки и продолжительность. Задачи



» Используя *Feng Office*, можно назначать задачи и управлять ими, отслеживать ход выполнения и время, отведенное на проект.

могут быть назначены клиентам и участникам Workspace. Также можно пользоваться Контрольной точкой [Milestone] из административного меню, дающей пользователям знать, когда этап заданий выполнен. Через Заметки пользователи могут обмениваться информацией между собой, взамен форума, блога или списка рассылки, при поддержке комментариев.

Точная настройка

Во вкладке отчетов *Feng Office* можно создавать отчеты по любым типам объектов. При помощи модуля Время можно задать временной интервал, и генерировать отчеты по времени, отображающие итоговое время для перечня заданий. Также можно получить отчет за любой временной промежуток и для любого пользователя.

Путем создания категории биллинга можно выставить время, которое каждый пользователь тратит на определенное задание, устанавливая биллинговую норму и присваивая ее пользователю. Странно, что американский доллар — базовая валюта системы, и нет возможности ее изменить. Еще ограничение: каждый пользователь может иметь только почасовой тариф.

В *Feng Office* есть организация документооборота и функция редактирования. Раз документ может редактироваться многими пользователями, есть способ проверить, открыт ли документ кем-то еще, т.е. нескольким пользователям одновременно редактировать не получится. И хотя можно отслеживать историю редактирования документа, сама эта возможность реализована не очень хорошо. Она показывает

изменения документа с привязкой по времени, но вместо того, чтобы показывать каждую правку, выводится только финальная версия. Для переключения на раннюю версию документа, нужно его загрузить, щелкнуть по ссылке Обновить файл [Update file] и скачать загруженную версию.

Еще одна интересная особенность *Feng Office* — создание шаблонов. Во многих организациях есть несколько рутинных этапов, иногда выполняемых регулярно — например, когда заводят нового клиента. Вместо того, чтобы создавать все задачи по новой, проще употребить шаблон.

Если вы не используете совместное ПО, то может показаться, что *Feng Office* — не вариант. Но благодаря подробной документации, вики и форумам вы оцените его многоликую природу и удобство в управлении группами и задачами. **LXF**



Свойства новскидку



Менеджер документов
Можно создавать, редактировать, сохранять и распределять любые тексты, с контролем версий.



Функции для профи
Добавлены функция массовой рассылки, диаграммы Ганта, поддержка мобильных устройств и не только.

LINUX FORMAT Вердикт

Feng Office CE 2.3.1

Разработчик: Feng Office
Сайт: www.fengoffice.com/web
Лицензия: Различные, ядро — лицензия BSD

Функциональность	7/10
Производительность	9/10
Простота использования	7/10
Документация	9/10

» Это прекрасный инструмент, коль скоро вы им владеете и хотите ловко управляться с группами пользователей и задачами.

Рейтинг 8/10

Fotoxx 13.09

Хочется лучше смотреться на фото, не разорившись на пластической хирургии? **Маянк Шарма** озирает редактор изображений, который все уладит.

Вкратце

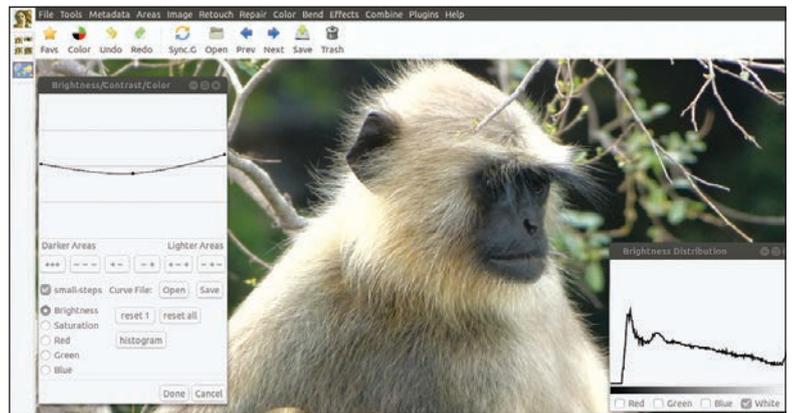
» Полноценное приложение для обработки изображений, в том числе RAW-файлов. См. также *GIMP* и *Raw Therapee*.

Для большинства пользователей ПК редактирование изображений ограничивается базовым кадрированием и автокорректировкой по одному щелчку мыши; эти функции есть в большинстве приложений работы с изображениями. *Fotoxx* отличается от других популярных приложений большей доступностью. Он не совсем для новичков: кое-какое знакомство с процессом редактирования тут потребуется. Но начинающие пользователи тоже могут вдохнуть жизнь в снимки: инструменты улучшения изображений сгруппированы по функциям, например, Ретушь [Retouch], Восстановление [Repair], Поворот [Bend] и т. д., что делает их понятнее и достижимее, чем бесконечные списки меню, как у аналогов.

Fotoxx доступен в репозиториях большинства настольных дистрибутивов, а разработчик на своем сайте поддерживает 32 и 64-битные Deb-пакеты для версий, наряду с tar-архивами и инструкциями по установке для дистрибутивов, основанных не на Debian.

Слева в приложении находятся вкладки для переключения от изображения ко всей галерее; в обоих видах есть собственные меню, которые можно расположить сверху или слева. При первом запуске приложения вам предложат поместить его в галерею. Когда изображения индексируются, управлять и выбрать любые изображения для редактирования можно будет одним щелчком.

Fotoxx полон инструментов для редактирования. Можно начать от простого: обрезка, масштабирование и поворот



» Перед использованием приложения рекомендуем пролистать руководство пользователя.

снимка, и перейти к более продвинутым хитростям, таким как изменение цветового баланса, уменьшение шумов и т. д. В последней версии появилась новая функция Mashup, позволяющая сделать монтаж, многократно добавляя изображения и тексты на виртуальный холст. При помощи меню Combine [Скомбинировать] можно создать HDR-изображение из нескольких снимков, числом до девяти. Приложение автоматически выравнивает и сочетает изображения, а после этого можно еще сделать настройки вручную. Точно так же для создания изображений с большой глубиной резкости можно сочетать изображения с различными настройками фокуса.

Полезные функции

Учтите, что сочетание снимков требует большого объема вычислений и может занять от минуты до получаса, в зависимости от размера и количества изображений и ресурсов оборудования. Еще одно из двух мест, где *Fotoxx* запинаяется — падает во время сочетания снимков камерой Canon DSLR с высоким разрешением.

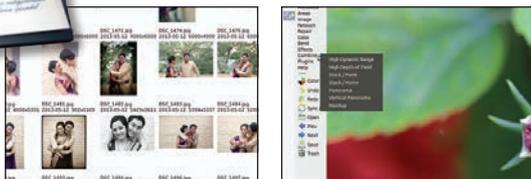
Другой скользкий момент — обработка RAW-файлов. Когда это происходит, *Fotoxx* сначала преобразует их в TIFF (16-разрядный цвет), а потом открывает TIFF для редактирования. Для преобразования группы изображений в редактируемые изображения TIFF (а также JPG и PNG) можно использовать утилиту Пакетного преобразования [Batch Convert Tool]. RAW-файлы от Canon обрабатывались превосходно, но у камеры Panasonic были искажены и отливали фиолетовым.

Начинающие фоторедакторы будут благодарны функции Voodoo Enhance из меню Retouch, автоматически улучшающей изображение. Опытным редакторам это меню пригодится для изменения яркости, контраста, насыщенности цвета и кривой цветового баланса. В меню Repair можно редактировать дефекты изображения. В меню для затирания лишних элементов изображения, скажем, соринку, есть полезная функция Smart Erase. Начиная с версии 13.05 это сглаживающая функция, которая поможет убрать угловатости в снимках с низким разрешением.

Fotoxx — прекрасное приложение для улучшения фотографий, с интуитивным интерфейсом и инструментами редактирования и для новичков, и для ветеранов, с надежной документацией; к тому же он активно развивается. **LXF**



Свойства навскидку



Менеджер фотографий

Fotoxx умеет и управлять изображениями, организовывать и находить изображения, по тегам и геотэгам.

Объединение картинок

Функции меню Combine пригодятся при создании HDR-изображений, панорам и многого другого.

LINUX FORMAT Вердикт

Fotoxx 13.09

Разработчик: Christoph Korn
Сайт: www.kornelox.com/fotoxx.html
Лицензия: GPL v3

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

» *Fotoxx* идеален для тех, кто хочет обрабатывать изображения быстро, но побойдается GIMP.

Рейтинг 8/10

Fuze

Fuze разработал настольный ПК на Raspberry Pi, который несколько обескуражил **Бена Эверарда**.

Вкратце

» Корпус Raspberry Pi со встроенной клавиатурой. См. также корпус PiBow с набором дополнений и комплект «для обзаведения» в Raspberry Pi — ModMyPi.

Когда Fuze появился в Башнях LXF, волна восхищения прокатилась по нашим офисным столам. Fuze взял наше любимое устройство (Raspberry Pi) и поместил в форму машинки, навсегда оставив след в наших сердцах (BBC Micro). Сходство с BBC Micro глубже, чем просто физическое. Машина загружается в модифицированную версию Raspbian, где основное добавление — Fuze Basic. Он не только похож синтаксисом на BBC Basic, но и выглядит как тот — брусковый белый шрифт на черном фоне. Хотя это придает ему некую ретро-крутость, мы не уверены, что такое подходит устройству для школ. Не обязательно навязывать детям уродливую устаревшую графическую среду. Нет подсветки синтаксиса, нет управления мышью и копирования, а вырезание и вставка ограничены.

Пожалуй, единственное, что украшает Fuze Basic — наличие документации. Машина поставляется с удобным Руководством программиста, где описаны все команды и дано несколько примеров. Но стоило нам начать программировать, как мы быстро осознали недочеты этого руководства: не зная имя команды, которую необходимо использовать, вы будете вынуждены листать страницы, пока не найдете нечто полезное. Там также нет примеров побольше или более углубленных инструкций. Сравним это, например, с Python — языком, рекомендованным Raspberry Pi Foundation. *Idle*, IDE по умолчанию, конечно, не идеальна, но на голову выше Fuze Basic: подсветка синтаксиса и управление мышью налицо, шрифты приятнее, и можно одновременно открывать



» В комплекте Fuze есть все для начала программирования.

больше одного файла. Функции не продвинутые, но полезные тем, кто только начал программировать. Более того, у него хорошая документация и хорошее присутствие в онлайн-поиске, при бесчисленном количестве статей, учебных пособий, книг и целой секции Stack Overflow специально про него. И Python доступен на многих видах компьютеров, не только на Raspberry Pi.

Слишком базово

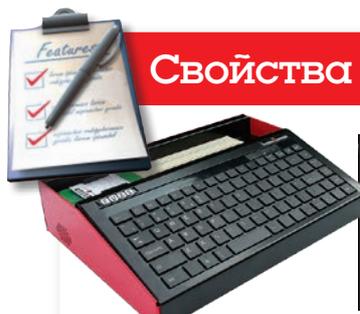
Увы, наши вопросы к Fuze не ограничиваются ПО. Клавиатура кажется хлипкой и не совсем приятна в использовании. Хуже того, она неестественно высока, и угол ее наклона неудобен для печатания. Прилагаемая мышь не лучше: она тоже не выглядит долгожительницей. Машинка эта, правда, не для работы днями напролет, но по нашим ощущениям, процесс печатания вызывает отторжение сразу же.

Корпус выглядит надежным и, пожалуй, протянет дольше всего из вышеперечисленного. Но в чем его плюсы по сравнению с обычными корпусами Raspberry Pi с внешними периферийными устройствами? Если честно, нам в голову ничего не пришло. Имейся тут хотя бы гнездо для велосипедного замка, это было бы хорошо для сохранности, но такового тоже нет. С другой стороны, корпус усложняет замену частей (скажем, клавиатуры) в случае поломки. Также нельзя добавить периферийное устройство — скажем,

камеру или платы расширения GPIO. Вдобавок упрямытие внутренностей машины как-то противоречит духу Raspberry Pi, который поощряет эксперименты, а не препятствует им.

Ну, по крайней мере Fuze продемонстрировала GPIO и макетные платы для построения простых схем: в комплекте есть LED, резистор, зуммер и 7-сегментный дисплей. Все это программируется в Fuze Basic, но синтаксис станет громоздким, стоит вам выйти за пределы Включение/Выключение света. Нечто вроде PiFace, объединенного с Python, предложило бы куда более богатую и расширяемую среду разработки для юных умов. **LXF**

Свойства навскидку



Корпус
Прочный и красочный, со встроенной клавиатурой, но неудобный.



Fuze Basic
Ностальгический взгляд назад, в былые времена невинности.

LINUX FORMAT Вердикт

Fuze
 Разработчик: Fuze
 Сайт: www.fuze.co.uk
 Цена: £7–180

Функциональность	5/10
Производительность	8/10
Простота использования	5/10
Документация	5/10

» Ретро-вид создает неверное представление о комплекте, который очень ограничен и в ПО, и по железу. Ну хоть корпус выглядит мило.

Рейтинг 5/10



МЕЗОН • ИНТЕГРАЦИЯ

Комплексные решения в области безопасности

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

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

Наши системы безопасности предназначены для:

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

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

ТелеВизард

Многофункциональный сервер общего назначения



ТелеВизард Авто

Самая малобюджетная и высокоэффективная система распознавания автомобильных номеров



Источники видеосигнала

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



si.mezon.ru

Тел. +7 812

309 0686

Тел./факс +7 812

640 4990



КИТКАТ ВМЕСТО KEY LIME PIE

Долгожданная новая версия

Анонсированы мобильная платформа Android 4.4 и смартфон Nexus 5.

31 октября компания Google анонсировала мобильную платформу Android 4.4 KitKat и представила новый смартфон Nexus 5 на ее основе. Android 4.4 сократил потребление ОЗУ: разработчики заявляют, что вместе с приложениями Google, в т. ч. браузером *Chrome*, он комфортно работает на устройствах с 512 МБ ОЗУ. Публикация кода Android 4.4 в открытом репозитории AOSP и выпуск обновлений прошивок на базе Android 4.4 ожидается для устройств Nexus 4, Nexus 7, Nexus 10, Samsung Galaxy S4 и HTC One Google Edition в пределах недель. Также анонсирован новый смартфон Nexus 5, с 4,95-дюймовым экраном 1920 × 1080 (445 ppi), 4-ядерным процессором Qualcomm Snapdragon 800 (2,3 ГГц), GPU Adreno 330 (450 МГц), 2 ГБ ОЗУ, 16- или 32-ГБ Flash, Wi-Fi 802.11

» Смена названия версии пошла Android на пользу — KitKat знают во всем мире.



a/b/g/n, Bluetooth 4.0, NFC, GPS, 4G LTE (опция), microUSB, аккумулятором 2300 мАч на 17 часов разговора по телефону и 8,5 —

навигации в Сети, 8- и 1,3-Мпикс камерой; размер устройства 69,17 × 137,84 × 8,59 мм, вес 130 г. Оно поступит в продажу по цене \$349 (16 ГБ Flash) и \$399 (32 ГБ Flash).

Внутренняя ветка Android, разрабатываемая в недрах Google, синхронизируется с общедоступным репозиторием, на базе которого формируются независимые сборки. Код уже доступен для загрузки через Git-репозиторий проекта.

Независимое сообщество CyanogenMod, развивающее альтернативную сборку Android, представило новую серию Milestone-сборок CyanogenMod 10.2 на базе Android 4.3, вполне пригодных для повседневного применения, но еще не получивших статус стабильных: готовы сборки для более чем 70 моделей устройств. Параллельно ведется работа по созданию версии на Android 4.4.

ПАТЕНТНЫЙ ТРОЛЛЬ АТАКУЕТ

Наезд на Android

Контролируемая Apple и Microsoft компания судится с Google.

В 2011 году шесть тысяч патентов, принадлежащих обанкротившейся корпорации Nortel, продали с аукциона за \$4,5 млрд. Google участвовала в аукционе, но не стала перебивать цену за этот набор патентов, приобретенный вновь созданной компанией Rockstar Bridco, соучредители которой — Microsoft, Apple, RIM, Ericsson, и, как ни странно, Sony (производитель планшетов и сотовых телефонов на платформе Android).



» Офис компании Nortel, которой принадлежал патентный портфель, теперь занимает компания-тролль.

Два года Rockstar разыскивала патенты, которые нарушает Google в своей мобильной ОС, и, наконец, против Google и семи производителей устройств — Asustek, HTC, Huawei, LG Electronics, Pantech, Samsung и ZTE — был подан иск. Google является ответчиком впервые: ранее патентные тролли, под предводительством Apple, Microsoft и Nokia, не атаковали Google напрямую. Иск подан в суд Восточного округа штата Техас, известный лояльным отношением к патентным троллям. Интересы истца представляет юридическая фирма McKool Smith, достигнувшая успешного завершения ряда крупных патентных разбирательств, в том числе против Apple, Microsoft и Qualcomm.

7 патентов, фигурирующих в иске, касаются технологий «ассоциативного движка для поиска», «машины, показывающей рекламу пользователю, который

ищет интересующую его информацию в Сети», «инструмента графического интерфейса пользователя, определяющего способ навигации по электронным документам», «фильтра интернет-протокола» и «интегрированного центра сообщений». Самый старый получен в 1997 году, еще до появления компании Google, а самый новый — в 2007-м. Еще в 2011 году Google предупредила, что патенты Nortel куплены ради целенаправленной агрессии против Android. Министерство юстиции США проверяло сделку Rockstar Bridco с Nortel, но не выявило опасности применения данных патентов для давления на конкурентов посредством судебных разбирательств. В прошлом году издание Wired выявило, что в Rockstar с полной занятостью трудятся 10 экспертов по обратному инжинирингу, разбирая различные продукты на предмет возможных нарушений.

СБОРНЫЙ СМАРТФОН

Своими руками, для себя

Модульные смартфоны врываются в нашу жизнь, как некогда сделали ПК.

Скоро можно будет собрать смартфон самим. По крайней мере, заменить его основные модули.

Компания Motorola представила новую платформу для создания модульных смартфонов — Project Ara, развиваемую в соответствии с принципами Open Hardware. Целью проекта является создание заменяемой аппаратной начинки для смартфонов, которая поможет сделать для оборудования то, что сделала Android в области программного обеспечения — сформировать независимую экосистему, понизить барьеры вхождения на рынок, способствовать развитию инноваций и сократить сроки разработки продукта.

Платформа позволит добиться более открытой и четкой взаимосвязи между пользователями, разработчиками и применяемыми ими смартфонами. Пользователь сам решит, какими функциями должен обладать его телефон, как он должен выглядеть, из чего он будет сделан, сколько будет стоить и как долго продлится его эксплуатация. Для воплощения подобной идеи в жизнь в рамках проекта Project Ara разработана модульная система, позволяющая компоновать смартфон из заменяемых комплектующих.

Project Ara состоит из двух базовых компонентов: эндоскелет и модули. Эндоскелет является структурным каркасом

для компоновки модулей. В качестве модуля могут быть оформлены любые элементы начинки смартфона, от процессорных блоков до экранов, клавиатур, камер, расширенных аккумуляторов и разнообразных сенсоров. Таким образом, выбирая модули на свой вкус, пользователь самостоятельно комплектует телефон, оптимально подходящий для его потребностей и финансовых возможностей. В любой момент уже используемый модуль может

быть заменен на более совершенный вариант или в телефон добавлена дополнительная функциональность.

Через несколько месяцев планируется организовать рассылку приглашений для начала разработки модулей. Зимой ожидается первый выпуск инструментария для разработчиков модулей — MDK (Module Developer Kit).

Помимо этого, Motorola заключила сделку с производителем 3D-принтеров, компанией 3D Systems. Ожидается, что именно на ее мощностях будут выпускаться компоненты нового смартфона.

Проект Ara на момент анонса выглядел совершенно невероятным. Сам факт того, что смартфон можно будет самостоятельно собрать вручную из отдельных деталей, поражает воображение. Что касается 3D Systems, то это, пожалуй, оптимальное решение для производства модульного смартфона. Возможности 3D-печати с каждым месяцем расширяются. Прямо сейчас компания собирает на Kickstarter средства для создания принтера, способного наносить проводящие дорожки на различные материалы: пластик, керамику и т.д. Компании 3D Systems придется несколько расширить производство и провести немало исследований, чтобы научиться изготавливать проводящие и функциональные материалы.

Открытость смартфона на аппаратном уровне также подразумевает и то, что другие производители могут выпускать свои модули для Ara. Motorola придется поразмыслить и над тем, как сделать производство компонентов максимально унифицированным.

Окончательная дата выхода смартфона в свет пока слишком далека и призрачна. Тем не менее, имеются данные, что разработчики получат в свое распоряжение первые прототипы для исследования уже этой зимой. **LXF**

«Пользователь сам решит, из чего будет сделан его смартфон.»

➤ Окончательная дата выхода смартфона в свет пока призрачна, но идея потрясающая.



Сравнение

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

Читалки новостей

Находить свежие новости, когда их выложат на сайтах, ниже достоинства **Шашанка Шармы**: эту работу за него делает программа.



Про наш тест...

Наш список — забавный микс из web-приложений и инструментов рабочего стола. Оба типа мы тестировали с помощью *Firefox* на рабочем столе *Xfce* и *Ubuntu 13.04 LTS*. Читалки RSS нельзя упрекнуть в жадности до ресурсов, но у нас было 2 ГБ ОЗУ на двухъядерном процессоре 1,86 ГГц, чтобы они остались довольны. Ищем мы, по сути, две основные функции: во-первых, стабильность приложения даже при работе с множеством новостных лент; во-вторых, эффективность представления всех ваших новостных лент. А в-третьих, важна поддержка мультимедиа, ведь Интернет давно уже не ограничен чисто текстовым контентом. Кроме того, мы ищем приложение, способное синхронизироваться с несколькими устройствами, поддерживающее совместное использование социальных сетей и достаточно гибкое для персональной настройки.

Наша подборка

- » Akregator
- » Feedly
- » InoReader
- » Liferea
- » RSSOwl

Really Simple Syndication (RSS) популярна среди разработчиков сайтов, но эта технология никогда не могла добиться такого уровня всеобщего признания, который большинство из нас инстинктивно ей приписывают.

В отличие от Facebook и Twitter, ваша бабушка и метрдотель вашего любимого ресторана, вероятно, и слухом не слышали об RSS или о программе для чтения. Однако только потому, что мир пренебрег RSS, сочтя каналы социальных сетей наилучшим способом узнавать последние новости и сплетни, RSS хуже не стал.

Программа для чтения новостей позволяет вам подписаться на ваши любимые сайты, и вам не придется регулярно

«Просматривайте сотни интернет-ресурсов из одного-единственного удобного приложения.»

их перелопачивать в поисках свежатинки: вы можете просматривать сотни любимых интернет-ресурсов из одного-единственного удобного приложения рабочего стола. Это — дар RSS. Более того, большинство программ для чтения RSS охотно загрузят не только традиционный текстовый контент, но и ваши любимые подкасты.

Многие клиенты электронной почты, скажем, *Thunderbird*, тоже могут служить читалками RSS, но подобные приложения мы в список не включили, потому что

занимаемся только специнструментами, чей хлеб — загрузка новостей.

В зависимости от числа новостных лент, загруженных вашим клиентом для вас, навигация по всей этой информации способна стать проблемой. Большинство программ чтения новостей решают ее, предоставляя расширенный поиск и фильтры. Вы также можете использовать пометки и цветовые коды, чтобы отмечать интересные новости, или раскладывать новости по папкам на основе разных параметров.

Управление новостной лентой

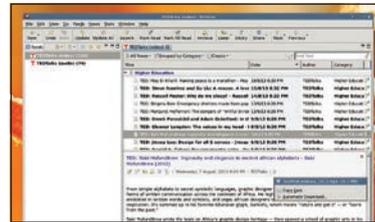
Новостей много не бывает?

Рбота читалки RSS заключается в скачивании новостей со всех ваших любимых сайтов. Но это лишь первый шаг. Куда важнее то, как именно собранная информация выдается вам. Когда надо читать сотни новостей с разных сайтов, в них легко запутаться. И тогда вы не сможете отслеживать те новости, которые уже прочли, те, которые пока не прочли, и важные новости, которые надо сохранить. С помощью *RSSOwl* можно создавать наборы новостей (или категории), чтобы ваши новости выглядели упорядоченно. Например, у вас может быть набор, где вы собираете комиксы *xkcd* и *Dilbert*. Или — подборка новостей, или отдельные наборы спортивных новостей, технических новостей и т.д. Более того, внутри каждой ленты можно создавать папки, что помогает управлению. Можно также создавать фильтры, чтобы новости автоматически сортировались и помещались в определенные папки. Например, вы можете создать фильтр для GPL по входящим юридическим новостям, и тогда все новости, касающиеся GPL, будут складываться в отдельную папку.

Akregator, приложение RSS в KDE, также позволяет сортировать входящие новости по папкам. А каждая папка может иметь подпапки. К сожалению, приложение не позволяет осуществлять поиск по всем лентам, а только по папке за раз.

Liferea тоже вполне стабильно работает со множеством новостных лент. Ее простой поиск позволяет находить ключевые слова по всем лентам, а уж расширенный поиск вообще превосходен, хотя и уступает *RSSOwl*. Вы можете задать несколько правил для поиска: например, статья, название статьи, статус, флаг, категория, название ленты, и т.д. Также *Liferea* позволяет сохранять результаты поиска в папке. Именно так вы фильтруете контент от входящих лент по специальным папкам.

Встроенный в *Feedly* поисковик по контенту можно применить для поиска контента по типу, например, технический (Tech), юридический (Legal), спортивный (Sports), и прочее. Он позволяет определять категории для входящих новостей и предоставляет простой интерфейс *drag-and-drop* для создания категорий или сортировки некатегоризированных лент



» *RSSOwl* нужно упростить процесс создания наборов лент. По этой части интерфейс не слишком интуитивен.

по имеющимся категориям. Здесь нет опции поиска по входящим новостям: чтобы получить доступ к этой роскоши, надо обновиться до премиум-версии. *Feedly* требует, чтобы ваш браузер поддерживал Flash, и предлагает три разных раскладки на выбор — *Cards* [Карточки], *Spotlight* [Главное] и *Headlines* [Заголовки].

Если вы готовы примириться с не похожим на других интерфейсом *InoReader*, то осознаете, что это отличное *web*-приложение для чтения новостей. Вы можете отмечать новости как любимые, и затем они окажутся в отдельной папке. Поиск весьма эффективен, но не позволяет выборочно указывать, искать ли только в названии или в самом тексте. Также можно ограничить поиск любой категорией.

Вердикт

- Akregator** ★★★★★
- Liferea** ★★★★★
- RSSOwl** ★★★★★
- Feedly** ★★★★★
- InoReader** ★★★★★

» Оба *web*-приложения идут с горячими клавишами, помогающими навигации по вашим лентам.

Производительность

Не заставляйте его величество ждать...

Вданном конкретном тесте мы оценивали два разных аспекта производительности читалок. Во-первых, на том, быстро ли каждая из них находила обновления ко всем лентам, и во-вторых, хорошо ли они работали при имеющихся ресурсах системы.

Вы могли бы сказать, что ресурсы — не такая уж проблема для многоядерной машины с несколькими гигабайтами ОЗУ, но не у всех людей дома стоит новейшее оборудование, а по работе новости читать приходится отнюдь не всем.

Программы для чтения RSS не особенно прожорливы в плане ресурсов, и, с учетом рода их деятельности, должны отлично работать на большинстве машин. Так зачем же мы тестируем производительность? Затем, что никому не нравятся приложения, которые зависают или обновляют новости так медленно, что завтрашние новости вы получите послезавтра. А то и медленнее!

Так вот, все приложения рабочего стола на нашем тестовом компьютере работали с исключительной быстротой. Они практически моментально загружались, и мы не ощущали никаких задержек, например, при переключении между новостными лентами или при изменении настроек.

Во всех наших читалках можно выборочно обновлять набор лент или папку по одному, или обновлять все ленты одновременно. Все настольные приложения выводят при обновлении лент прогресс-индикатор.

Ни одно из настольных приложений не проявило излишней алчности к системным ресурсам. FAQ *RSSOwl* дает понять, что раньше это, возможно, и было проблемой, но теперь уже нет. Мы даже пробовали изменить размер окна, пока приложения занимались скачиванием новостной ленты. Похоже, приложения были ничуть не против, и продолжали работать без запинки. Когда мы прошлый раз тестировали *Liferea*, это было серьезной проблемой, но с тех пор она оптимизировала свою производительность и привела себя в порядок. *Web*-приложения, особенно *InoReader*, казались несколько помедленнее. Тем не менее *Feedly* был достаточно оперативен, и обновлял ленты почти что мгновенно.



» Если вы не разворачиваете браузер на весь экран, растяните его горизонтально, чтобы видеть боковую панель.

Вердикт

- Akregator** ★★★★★
- Liferea** ★★★★★
- RSSOwl** ★★★★★
- Feedly** ★★★★★
- InoReader** ★★★★★

» *Web*-приложения работают не очень стабильно при обработке видеофайлов.

Удобство использования

Хорошо ли они делают свое дело?

Читалки новостей получают ленты с сотен разных сайтов, и навигация по этому приложению и необходимость как-то разбираться с этими данными может стать кошмаром. Графический интерфейс, отображающий всю собранную информацию логичным и понятным способом, является крайне важным моментом

хорошо продуманной читалки RSS. Итак, каждый из интерфейсов должен предоставлять пользователю возможность легко и быстро перемещаться с ленты на ленту и помогать сразу определить важные новости. Читалка также должна предлагать несложный способ хранения и просмотра новостей позднее. И, наконец, мы ищем инструмент,

способный помочь принять разумные решения по собранным нами лентам контента, не отнимая у нас слишком много времени или энергии.

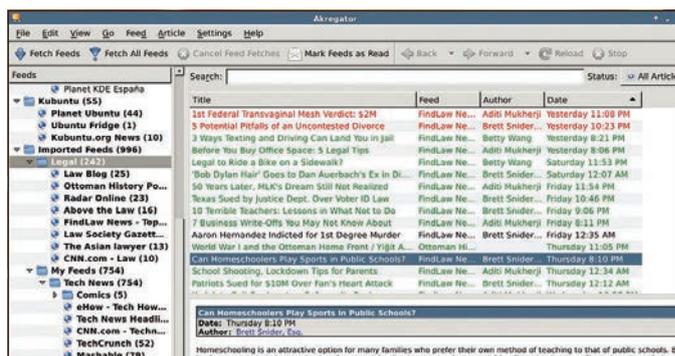
В этом разделе мы положимся на проверенную годами формулу: приложение, умудрившееся поднимать всю нагруженную на него тяжесть и при этом выглядеть мило, и будет чемпионом.

Akregator ★★★★★

Интерфейс по умолчанию Akregator похож на интерфейсы других настольных приложений рабочего Сравнения. Его можно сделать полноэкранным или выбрать комбинированную раскладку из меню View вверху.

Как и в Liferea, если вам нужно нечто сверх минимальной вкладки в новость, придется менять настройки. Щелкните правой кнопкой по левой панели и щелкните по Edit Feed [Редактировать ленту]. Перейдите во вкладку Advanced [Дополнительно] и выберите окошко Load the full website when reading articles [Загружать весь сайт при чтении статей].

Если вы этого не сделаете, то получите всего по паре первых строк каждой новости и ссылку Читать полностью [Complete story]. Нажатие на ссылку откроет новость во вкладке в Akregator. Настройки по умолчанию будут идеальными для тех, кто ограничен очень суровой политикой честного использования.



► Akregator — часть пакета KTorrent в KDE.



► После некоторых вложений, Feedly добавила учетную запись Pro за \$5 в месяц, где разблокирован поиск по лентам новостей.

Feedly ★★★★★

Интерфейс Feedly сильно отличается от настольных приложений. Ленты перечислены слева, что вполне стандартно, но в основной области отображения есть ряд опций отображения, таких как Magazine [Журнал] и Cards, которые используют сетчатый шаблон и стараются извлечь побольше изображений. Feedly предлагает и вид Title Only — стандартный список заголовков новостей, где нет уродливых белых квадратов, имеющих в других видах отображения, если новости нечего предложить для показа.

Новости в Feedly не отмечаются как прочитанные, пока вы не щелкнете по ним, что не даст вам никакой добавочной информации, кроме нескольких опций поделиться новостью.

Левая панель по большей части скрыта, и становится видна, когда вы перемещаете мышь в левую часть экрана. Это очень раздражает, потому что при каждом случайном пинке мыши влево вылезает боковая панель.

Синхронизация

Ладят ли они с другими устройствами?

Порыв узнать, что творится в мире, может накатить в любую минуту, а не только при бессоннице; часто это происходит, когда вы вне дома, но не в силах ждать момента, когда вы удобно устроитесь перед своим рабочим ПК. И что тут делать пользователю, жаждущему узнать новости? Помимо ввода вас в курс последних событий, читалки RSS должны также быть синхронизированы по всем вашим устройствам, иначе будет невозможно отслеживать, какие новости вы уже прочли, а какие — еще нет.

У Feedly и InoReader, ввиду их сетевой природы, здесь явное преимущество. Все, что вам нужно для получения доступа к этим web-приложениям — соединение с Интернетом. Все настройки, ленты и папки привязаны к вашей учетной записи. В результате вы всегда получаете новостные ленты именно в том виде, в котором они вам нужны, независимо от того, какое устройство используете. Более того, оба web-приложения в нашем Сравнении предоставляют приложения Android, что дает дополнительные функции. Feedly

также предоставляет приложения для устройств iOS.

Являя собой разительный контраст, все настольные читалки RSS лишены средств синхронизации. Максимум, на что вы можете надеяться — экспорт ваших лент с этих инструментов и импорт на другую машину. Liferea раньше позволяла синхронизировать разные устройства через Google Reader (пока Google не прикрыл его), но остальные приложения рабочего стола для чтения являются весьма обособленными.

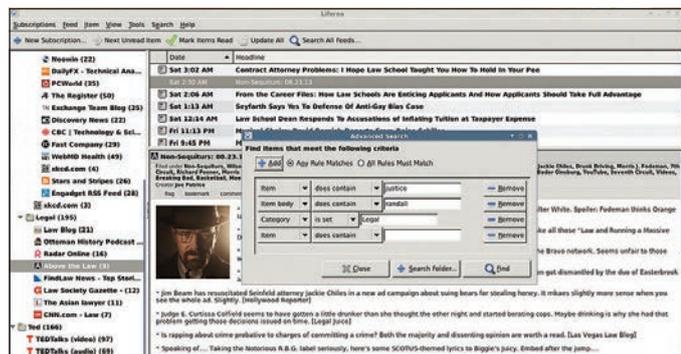
Вердикт

- Feedly ★★★★★
 - InoReader ★★★★★
 - Akregator ★★★★★
 - Liferea ★★★★★
 - RSSOwl ★★★★★
- » Отсутствие поддержки устройств Android простиительно, но отсутствие опции синхронизации — нет.

InoReader ★★★★★

Одна из наилучших черт InoReader в том, что он значительно облегчает импорт и экспорт лент. Эта функция традиционно предоставлялась только настольными приложениями, в том числе и тремя фигурантами нашего Сравнения, но InoReader позволяет импортировать и файлы OPML [Outline Processor Markup Language — язык разметки структуры на базе формата XML, — прим. ред.] тоже, так что если понадобится мобильность, очень легко переключиться с приложения рабочего стола: нажатием на кнопку Quick Menu в правом верхнем углу окна браузера, и выбором Import/Export из списка.

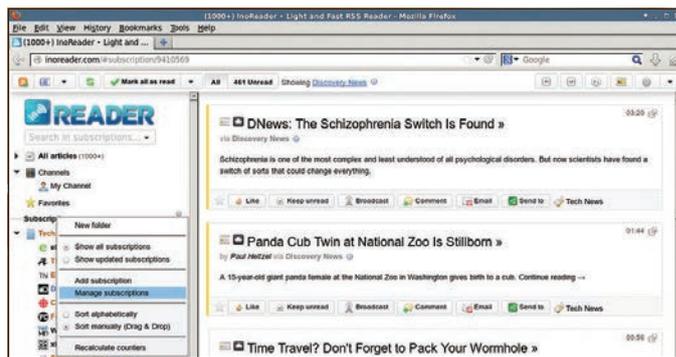
В отличие от Feedly, интерфейс очень четкий, но не предназначен для просмотра всего контента. Щелкните по маленькой кнопке слева от заголовка новости, чтобы загрузить всю статью, и вы увидите, почему. Или можете использовать кнопку рядом с ней, чтобы загрузить очищенную, текстовую версию статьи, которая предоставляется Readability (www.readability.com).



» **Liferea** поддерживает поиск и возможность воспроизводить подкасты.

RSSOwl ★★★★★

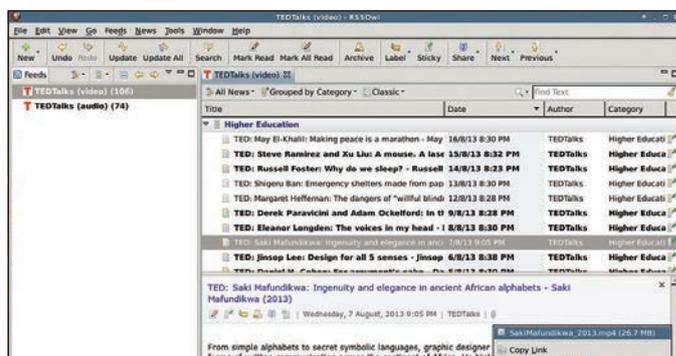
Интерфейс по умолчанию настроен на вид Classic, чего для большинства пользователей достаточно. Вы также можете выбирать между Newspaper [Газета], Headline и парой других видов отображения, но в них придется пожертвовать частью функций. Также RSSOwl выбрал левую панель для переключения по лентам новостей и их наборам. Каждый набор конкретного вида новостей определяется как новостная подборка: у вас есть Tech News [Технические Новости], Legal News [Юридические Новости], Sports News [Спортивные Новости] и т.д. Для перехода к другой подборке новостей вы щелкаете по маленькой, почти невидимой стрелочке вверху в левой панели, однако всплывающие подсказки при наведении мыши весьма удобны и сообщают о свойствах каждой кнопки. При добавлении новой ленты приложение не дает определить новую подборку новостей: сперва создайте подборку новостей, а потом добавляйте новый контент. Что нам показалось весьма удобным.



» InoReader позволяет импортировать ленты через файлы OPML.

Liferea ★★★★★★

Как и классический вид RSSOwl, интерфейс Liferea напоминает типичный почтовый клиент. Все ленты перечислены в левой боковой панели. С настройками по умолчанию вы увидите всего несколько строк текста основной части статьи. Дело в том, что Liferea не использует настроенный браузер по умолчанию. Щелкните правой кнопкой по каждой ленте на левой панели, и щелкните по Properties [Свойства]. Перейдите на вкладку Advanced и выберите Auto-Load Item In The Configured Browser [Автозагрузка статьи в настроенном браузере], и тогда контент отобразится нормально. Настроить так все ленты сразу нельзя, и придется повторять этот шаг для каждой новостной ленты в списке. Когда вы настроите Liferea на отображение контента, придется обратить внимание еще на одну вещь: горизонтальные полосы прокрутки. Если только вы не работаете с приложением в полноэкранном режиме [Fullscreen], надо будет изменить размер вручную, чтобы избавиться от этих полос.



» Многочисленные вкладки в RSSOwl умеют отображать ленты новостей бок о бок; а если вы не любите вкладки, просто отключите эту функцию.

Общение

Ну всем же нужно с кем-то общаться.

Помимо тех, кто обожает грузить всех подряд мельчайшими деталями своей прогулки по универсаму и закупок на ужин, большинство людей делится новостями, которые им кажутся достойными внимания других.

В Feedly надо щелкнуть по новости в ленте, чтобы получить доступ к возможности поделиться этой новостью. Вы увидите кнопки для Twitter, LinkedIn, Facebook, Google+, и кнопку отправки новости друзьям электронной почтой. Нажатие на кнопку электронной почты запустит мастер

настройки почтового клиента по умолчанию вашего дистрибутива, если он еще не настроен. В InoReader нужно щелкнуть кнопку Send to [Отправить] внизу каждой новости, чтобы получить доступ к кнопкам для четырех упомянутых популярных социальных сетей и множества других, например, Delicious, Tumblr, StumbleUpon и т.д. В отличие от Feedly, которая, чтобы поделиться новостью, вынуждает вас использовать только настольный почтовый клиент, InoReader позволяет рассылать новости через Gmail прямо из браузера.

Запустив Akregator в KDE, можно делиться новостями с другими по электронной почте, но это — расширение его функции выдачи общего доступа. В других же рабочих столах нельзя даже этого. RSSOwl поддерживает ряд сетей, но не все они активированы. Щелкните по Edit > Preferences > Sharing [Редактировать > Параметры > Выдача в доступ], чтобы настроить желаемые. Liferea дает делиться новостями только в одной настроенной сети за раз. Когда вы делаете на новости закладку, она посылается в указанный вами web-сервис.

Вердикт

- Feedly ★★★★★
 - InoReader ★★★★★
 - RSSOwl ★★★★★
 - Liferea ★★★★★
 - Akregator ★★★★★
- » В Liferea вы даже не можете поделиться новостью в нескольких сетях.

Поддержка мультимедиа

Где мои файлы с музычкой?

Вам интересна реакция разных читателей на подкасты и водкасты? Нам тоже, и мы обратились к подкасту TuxRadar, чтобы посмотреть, как они работают с файлами MP3 и Ogg. TuxRadar не делает vblog (пока), поэтому мы взяли интересное всем: Ted Talks [видеоконференция по технологиям], а Sapling Foundation любезно предоставил нам ленты RSS и для аудио-, и для видеонОВОСТЕЙ.

RSSOwl, на первый взгляд, даже не распознает файлы мультимедиа в той ленте, которую читает. Вы видите только текст новости, и ничего больше. Однако более пристальный взгляд позволил нам

разглядеть, что RSSOwl считает аудио- и видеофайлы вложениями. И если изображения отображаются в самом сообщении, то файлы мультимедиа передаются маленькой кнопке под заголовком. И у вас есть только одна возможность: скачать их. Второе наше настольное приложение ушло недалеко. Akregator тоже не воспроизводит файлов мультимедиа, но явно описывает эти файлы как Enclosure [Вложение]. Когда вы щелкаете по вложению, будь то аудио или видео, вам или дается возможность скачать его, или на файл указывается плееру, способному его обработать. Как и в RSSOwl, здесь

нет проблем с обработкой изображений: они показываются внутри, во встроенном браузере. Подражая другим настольным приложениям, Liferea требует для мультимедиа-файлов внешний плеер. Она тоже распознает файлы мультимедиа как вложения, но, в отличие от Akregator, вы можете только их скачать. Файлы мультимедиа перечислены как вложения внизу экрана. Если вы не знаете наверняка, что в новости есть мультимедиа, есть шанс просто не заметить маленький выпадающий список приложений внизу экрана.

Web-приложения в этом смысле куда лучше: они балуют нас встроенной поддержкой файлов мультимедиа. Вы можете слушать подкасты и смотреть видео в браузере. Feedly — чистый победитель, поскольку без труда справляется с обработкой файлов MP3, OGG и MP4. Однако его работа не слишком ровная: так, он не сумел воспроизвести все видео TED (файлы MP4). InoReader с аудиофайлами работает безупречно. Но когда мы указали ему на видеофайл, он стал спотыкаться. В наших тестовых попытках просмотра TED-видео InoReader буквально ползал. В конце концов воспроизведение пошло, но без всякой возможности управления из интерфейса, то есть установки на паузу или остановки. Кроме того, он полностью захватил браузер, лишив нас доступа к другим вкладкам.



➤ Для воспроизведения медиа-файлов в Liferea требуется сторонний плеер.

Вердикт

Liferea ★★★★★
 Feedly ★★★★★
 Akregator ★★★★★
 InoReader ★★★★★
 RSSOwl ★★★★★

» Liferea набирает наибольшее количество очков благодаря разделу Adaptability.

Настройки по умолчанию

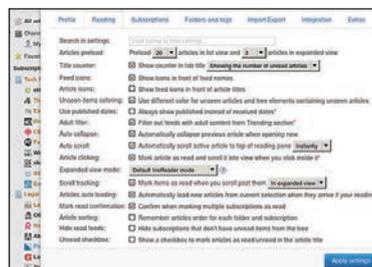
Стоит ли с ними возиться?

Настройки по умолчанию тестируемых нами читалок RSS должны удовлетворить большинство пользователей. Но все они позволяют производить целый ряд изменений. Помимо внешнего вида и прочих настроек по части лент новостей — например, частоты обновления ленты, времени хранения и т.д. — есть также ряд других, тоже изменяемых; обязательно просмотрите все имеющиеся настройки, иначе легко пропустить важную функцию.

Все приложения предоставляют основные функции, подходящие программе для чтения новостей. Но RSSOwl имеет также ряд доступных расширений, за которые вы его еще больше полюбите. Перейдите в Tools > Add-ons > Find Add-ons

[Инструменты > Расширения > Найти расширение] для добавочных функций. Например, если вы любите черкнуть небольшое примечание рядом с важной новостью в ваших лентах, вам стоит установить расширение Note. Чтобы изменить другие настройки, щелкните по Edit > Preferences. Вы можете изменить цвет, шрифт, определить маркировку, установить горячие клавиши на клавиатуре и прочие важные для ленты новостей настройки.

А есть читалки, которые прямо заставляют вас возиться с настройками для достижения оптимального эффекта. Liferea называет каждую ленту подпиской, и все подписки перечислены в левой панели. Чтобы Liferea воспроизводила файлы мультимедиа, надо изменить настройки



➤ InoReader предоставляет массу определяемых пользователем настроек.

для каждой подписки, а не применять глобальную настройку. Итак, щелкните правой кнопкой по Subscription на боковой панели, затем по Properties. Во вкладке Advanced щелкните по пункту Auto-load в настроенном браузере. Внутри Liferea появится мультимедиа-плеер для воспроизведения аудио- и видеофайлов.

Вердикт

InoReader ★★★★★
 Liferea ★★★★★
 RSSOwl ★★★★★
 Akregator ★★★★★
 Feedly ★★★★★

» Feedly предлагает множество разных тем, но этого мало, чтобы заслужить 4 звезды.

Читалки новостей

Вердикт

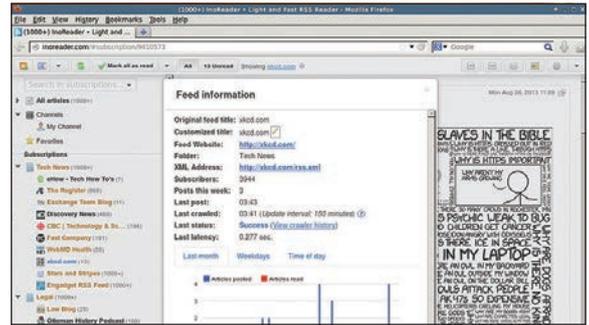
В активной разработке находится множество проектов RSS, и в их списке — множество настольных приложений, несмотря на тенденцию перехода на облачные сервисы. Мы более чем желали бы в таком случае порекомендовать приложение рабочего стола в качестве лучшей программы для чтения RSS, но в итоге все обернулось соревнованием между двумя web-приложениями.

InoReader есть в чем улучшить, но его импорт/экспорт лент новостей — функция превосходная. Связь этой программы с Readability также позволяет без всяких помех работать с контентом внутри программы, что трудно превзойти.

Feedly, самая рекомендуемая альтернатива Google Reader, должна приглядывать за InoReader и удвоить усилия, если надеется отыграть корону.

Хотя web-приложения предлагают несколько уникальных функций — приложения для мобильных устройств и работы

в облаке, но сообщения о смерти настольных приложений несколько преждевременны. RSSOwl уступает только двум web-приложениям, из-за отсутствия механизма синхронизации лент на разных устройствах. Если эта функция будет реализована, лучшее в облачных приложениях станет причиной того, что их станут избегать. Если произошедшее с Google Reader хоть что-то значит, то трудно сказать, когда популярное web-приложение может прикрыть лавочку. И Liferea, и Akregator требуют некоторой отладки, чтобы они смогли обрабатывать контент внутри. Некоторые могут счесть это хорошим примером самострела, но мы с этим не согласны. Эта функция обретет смысл, если вы вспомните, что многие интернет-провайдеры проводят нереалистичную политику честного использования для своих пользователей. В этом



случае никто не захочет бездумно скачивать по умолчанию полный контент для каждой новости, по которой щелкает.

Пусть Akregator не сумел реализовать синхронизацию лент на нескольких устройствах, с учетом того, как он связан с пакетом KDE Kontact, но если оставить это в стороне, то с самим приложением все нормально. Если вы не хотите делиться новостями с другими и вам нужно приложение KDE, способное просто и быстро находить для вас ленты новостей, почему бы не взять и Akregator.

» InoReader может стать только лучше из своего нынешнего состояния.

«InoReader есть в чем улучшить, но его импорт/экспорт лент превосходен.»

I InoReader ★★★★★
 Сайт: www.inoreader.com Лицензия: Проприетарная Версия: нет
 » Прекрасна в работе и имеет все функции для чтения новостей.

IV Liferea ★★★☆☆
 Сайт: www.lzone.de/liferea Лицензия: GPL Версия: 1.8
 » Нам нравится Liferea, хотя подиума ей не взять.

II Feedly ★★★★★
 Сайт: www.feedly.com Лицензия: Проприетарная Версия: 17.2
 » Уступает в гонке только из-за отсутствия некоторых функций.

V Akregator ★★★★★
 Сайт: www.kontact.org Лицензия: GPL Версия: 1.6
 » Akregator вовсе не плох, но другие предлагают больше.

III RSSOwl ★★★★★
 Сайт: www.rssowl.org Лицензия: Eclipse Public License Версия: 2.2
 » Одно из самых многообещающих настольных приложений.

Обратная связь
 А какая ваша любимая читалка RSS? Или вы работаете над своей? Поделитесь с нами своими мнениями на lx.f.letters@futurenet.co.uk.

Рассмотрите также...

Мы вообще-то хотели включить в наше сравнение Newsblur. Судя только по бесплатной демо-версии, она может с легкостью победить все прочие читалки. Увы, лучшее в ней — эта самая демо-версия. Когда вы регистрируетесь, перед вами появится экран, сообщающий, что вы стоите в очереди вместе с сотнями других, ожидая свободной учетной записи. Мы оказались в очереди

под номером 548. Но если вы хотите заплатить, то владельцы премиум-аккаунта получают полный доступ, и немедленно.

Есть множество онлайн-сервисов, более чем готовых занять место Google Reader. Если вы не против запустить собственный сервер RSS, можете прибегнуть к *Tiny Tiny RSS*. Это даст гарантию, что ваши данные будут действительно

вашими. Хотя настройка собственного сервера RSS может показаться делом трудоемким, имеющиеся свидетельства и обсуждения онлайн говорят о том, что это вполне посылно.

Для продвинутых пользователей есть *Newsbeuter*, читалка командной строки. Проект активно разрабатывается, и новая версия появилась в начале этого года. **LXF**

МОБИЛЬНЫЙ LINUX



Откройте новое поколение мобильных ОС с открытым кодом с **Беном Эверардом**.



Точную дату назвать трудно, но в некий момент последних двух лет Linux стал основной операционной системой для смартфонов, а сообщения последней пары месяцев показывают, что то же происходит с планшетами. И все это благодаря Android. Однако Android не похож на настоящий Linux. Изначально ядро отпочковалось от Linux (хотя сейчас они переживают слияние), и то, что с ним используются компоненты с закрытым кодом, оставляет во рту многих сторонников открытого кода кислый привкус.

Мир Linux не просто наблюдал за битвой техно-титанов, сложа руки. Разнообразные компании и организации в поте лица трудились

над разработкой новых мобильных операционных систем на Linux.

Некоторые из этих ОС основаны на новых идеях и концепциях, а некоторые являются программируемыми открытыми версиями существующих систем. Можно уверенно утверждать, что неко-

«Мир Linux не просто наблюдал за битвой техно-титанов.»

торые из рассмотренных здесь систем почтуют с миром, не оставив особого следа, но при правильном сочетании технической грамотности,

талантливого дизайна и деловой сметки одна из них вполне может штурмом взять мир и ознаменовать новую эру мобильного Linux.

Оборотная сторона мобильного рынка в том, что устанавливать новые операционные системы на такие устройства намного сложнее, чем на десктопы. Здесь меньше стандартов, и новые системы обычно поддерживают не все устройства. Например, старый Nexus 7 — один из самых популярных, и является тестовой системой в нашей статье. На момент написания только одна из систем есть в магазинах (Firefox OS), хотя с учетом темпов, с которыми развивается мир мобильных телефонов, в тот момент, когда вы будете читать эту статью, возможно, их будет доступно больше.

CYANOGENMOD

Это Android, мы теперь знаем.

CyanogenMod создан с помощью Android Open Source Project, и является свободной ОС, которая очень похожа (с точки зрения пользователя) на версию Google.

Мы настроились на превращение нашего верного Nexus 7 в устройство чисто с открытым кодом, чтобы рассмотреть его отличия от обычной смеси открытого кода и проприетарных приложений. Установка оказалась простой; нужно было просто проверить, установлены ли нужные инструменты (в системах на базе Debian и Ubuntu — пакет *android-tools-fastboot*), а затем следовать инструкциям с www.cyanogenmod.org для нашего устройства.

Результат сильно напомнил вид нашего устройства, когда оно только-только прибыло к нам с Mountain View. Даже обоим отличались знакомым шиком Android 4. Здесь была пара других опций, позволявших индивидуально настроить устройство под свои предпочтения, но за полгода использования Nexus автор полюбил Android по умолчанию.

Хотя ОС и похожа, но ни одно из приложений Google не идет с CyanogenMod. Более того, здесь нет Google Play, поэтому мы не смогли установить их. По сути, это предопределялось нашим решением: установить Google Play на CyanogenMod вполне возможно, но это было бы против правил создания среды с открытым кодом на нашем Nexus 7.

Для начала требовался достойный web-браузер. Честно говоря, без него наш планшет стал бы для нас почти бесполезным.

Основной браузер, идущий с CyanogenMod, страшен, как смертный грех, и довольно неприятен в использовании. На Android мы бы использовали *Chrome*, но чтобы соблюсти принцип открытости кода, выбрали *Firefox*. На <https://ftp.mozilla.org/pub/mozilla.org/mobile/releases/latest/android/multi> имеется файл установки. И так, мы получили на нашем планшете первоклассный мобильный браузер. Даже если больше ничего не работало бы, компьютерное устройство было бы вполне приемлемо. Однако, к счастью, это еще не конец. Нам не пришлось отказаться от магазинов приложений лишь из-за отсутствия Google Play. Есть F-Droid (<https://f-droid.org>) — рынок свободных программ для Android и CyanogenMod. Просто скачайте файл установки с их сайта, и вы получите множество программ с открытым кодом для своего устройства. Запасы Google, конечно, богаче, и если вам нужна тысяча разных способов просматривать картинки с котятми, вы, вероятно, разочаруетесь; зато F-Droid содержит приложения для всех основных потребностей (пользователи Android, обратите внимание: F-Droid — отличное место для поиска приложений без рекламы, даже если вы не готовы полностью перейти на открытый код).

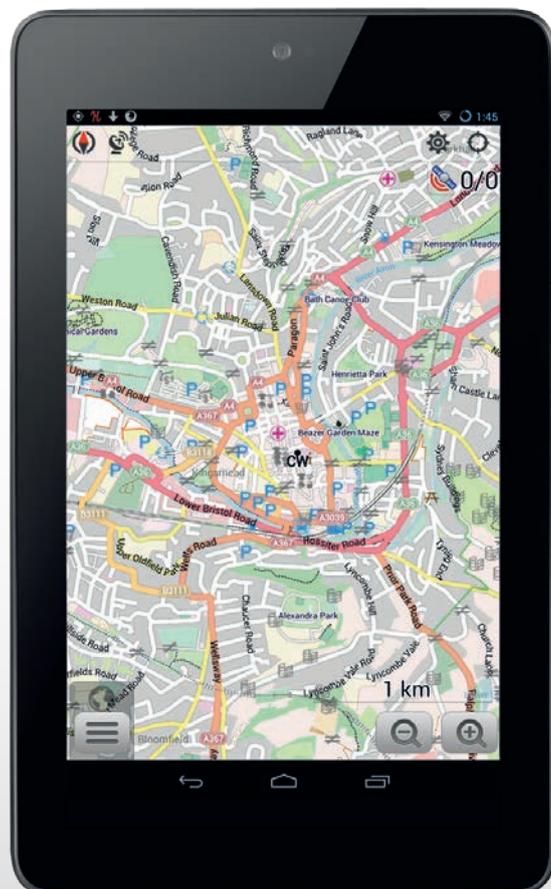
Планшеты все используют по-своему, но нам важен доступ к файлам в облаке и календарю. Ради них мы уже переключились с Google на *OwnCloud*, и смогли получить доступ к ним из браузера. Однако мы нашли *aCal* и *OwnCloud* в F-Droid, которые тоже позволили нам это сделать. *OSMAnd* вступил в игру в качестве картографической программы, и предложил нам просмотр карт в режиме оффлайн, который Google удалил из своих продуктов.

Основные приложения электронной почты и музыкальные приложения работали отлично, и мы вполне подготовились к продуктивной работе. Однако непроизводительное применение нашего планшета оказалось довольно бледным. На F-Droid есть несколько игр, хотя их количество не назовешь чрезмерным. Установив *Firefox*, мы также получили доступ к Firefox Marketplace (<https://marketplace.firefox.com/search>), где можно установить приложения, в том числе и игры. Оно открывает большие возможности, и, надо надеяться, с ростом популярности Firefox OS здесь будет появляться все больше приложений. Однако, в отличие от F-Droid, не все, что вы здесь найдете, будет с открытым кодом.

В этом пункте мы позволили себе поступиться принципами. Можете считать это прагматизмом или слабостью, в зависимости от вашей точки зрения. За несколько лет мы приобрели много любимых игр для Android через Humble Bundle, и их можно установить на CyanogenMod точно так же, как и на Android. И хотя код их закрыт, они свободны от DRM, и приобретая их, мы поддерживали EFF, чего для нас и достаточно. На этом наш свободный планшет был полностью настроен, и каждый его бит был так же полезен, как в Google Android.



► Кроме отсутствия сервисов Google, большинство пользователей даже не заметят разницы между CyanogenMod и Android.



► *OSMAnd* хранит все карты на вашем устройстве, хотя это может занять сотни мегабайт.

«Мы вполне подготовились к продуктивной работе.»

UBUNTU TOUCH

Донесем спорный рабочий стол до телефонов.

С самого появления Unity считался предназначенным для тачскринов, и вряд ли кого-то удивило объявление Canonical о запуске специальной версии Android для тачскринов. Конечно, не все ожидали специальной версии для телефонов. Невзирая на провал кампании по сбору средств на Ubuntu Edge, Canonical по-прежнему надеется запустить телефон в первом квартале будущего года, и в этом деле у них уже подбралась впечатляющая компания партнеров.

При первом запуске вы либо вздохнете с облегчением, либо сильно расстроитесь: на планшете нет того же самого рабочего стола Unity, которым столь (печально) знаменит Ubuntu. Пользовательский интерфейс Touch, однако, явно вдохновлялся этой средой рабочего стола. И при первом же запуске сразу становится

«Сразу становится ясно, что это — ранний релиз разработки.»

ясно, что это — ранний релиз разработки.

Здесь множество неловкостей, недо-реализованных функций и прочих досадных штук. Например,

энергопотребление настолько велико, что Nexus 7 теряет заряд, еще будучи подключенным к розетке и при весьма умеренном использовании. И вас это быстро начнет раздражать, если вы используете свой планшет вдали от источника питания. Зато ваши руки не замерзнут — скорее, поджарятся.

Чтобы дать вам пример текущего состояния разработки, мы попытались воспроизвести на нем музыку. Первой проблемой оказалась загрузка музыки на планшет. Ubuntu Touch не поддерживает никаких протоколов данных USB, и мы не смогли воткнуть провод и передать данные с компьютера. Тогда мы попытались скачать их с нашего резервного сервера *OwnCloud*, но JavaScript в *Owncloud* поломал браузер. В конце концов нам удалось с помощью инструментов разработки Android открыть оболочку на планшете (**adb shell**), и установить пакет *openssh-server*, а затем передать наши данные через SFTP (имя пользователя/пароль — *rphablet/rphablet*). Добившись этого, мы обнаружили, что плеер не воспроизводит файлы MP3. Тут уж нет смысла жаловаться, что на ранней стадии он пока не закончен.

Естественно, не закончен. Ну и ладно, вместо этого мы рассмотрим, что получится извлечь из ОС.

Ubuntu Touch сконцентрирован вокруг *Scopes*. Они работают более или менее так же, как на рабочем столе Unity. Их цель — дать пользователю обзор не только того, что имеется на его планшете, но и всей цифровой вселенной. Стандартные *Scopes*



Ubuntu Touch Launcher означает, что все ваши любимые приложения всего в одном движении от вас.

позволяют искать коммерческую музыку и видео, а также просматривать свои приложения и домашний экран. Это прекрасно, чтобы вы ощутили, для кого будет работать система (и, со временем, как отлично выглядеть), но отнюдь не годно для финального продукта. А ведь на телефоне это куда важнее, чем на настольной системе. Пока что создается ощущение, что *Scopes* существуют единственно как возможность продавать пользователю контент (музыку и фильмы), а вовсе не для помощи в его действиях. Надеемся, что это будет исправлено до первого релиза.

Именно ощущение, производимое пользовательским интерфейсом, действительно отличает Ubuntu Touch. Большая часть значков управления появляется, если провести с края экрана к середине. Пользователям Android 4 это будет знакомо, потому что так запускается Google Now. Мазните по экрану сверху, и вы увидите меню *Settings*; справа — появится *Launcher*, снизу — меню *Current Apps*, и надо двигаться вправо, чтобы переключаться между запущенными приложениями. Последним жестом весьма удобно управлять окнами: практически создается ощущение перелистывания страниц книги. Однако нам все эти движения казались довольно нелепыми, пока мы не вытащили планшет из чехла.

Вот, собственно, и все про Ubuntu Touch. Возможно, вы обольщались, что мы расскажем вам о магазине приложений, но на данный момент его нет. Вы можете установить добавочные программы, войдя туда со своего компьютера через **adb shell** и затем используя знакомую **apt-get**.



Раздел *Featured in Video* и *Music scopes* предлагает ряд обложек. Правда, хотелось бы, чтобы они относились к нашим альбомам, а не к тем, которые Canonical пытается нам продать.

FIREFOX OS

ОС на web-технологиях для мобильного телефона.

Mozilla вспоминается отнюдь не первой, когда речь заходит об организациях из индустрии мобильных телефонов — по крайней мере, не как разработчик ОС, хотя это могло бы быть мудрым ходом со стороны разработчиков

браузера. В последние несколько лет имел место отток к браузерам на базе *WebKit* (например, *Chrome* и *Safari*); но и утратив долю рынка настольных систем, *Firefox* остается главным игроком. Правда, в мобильном мире его едва ли считают серьезным конкурентом. Он доступен под *Android*, но его доля на рынке настолько мала, что ее даже трудно учесть. Если модное утверждение о смерти настольных ПК верно, то и *Mozilla* тоже попадает в затруднительную ситуацию, разве что ей удастся развернуть в свою пользу ситуацию на мобильном рынке.

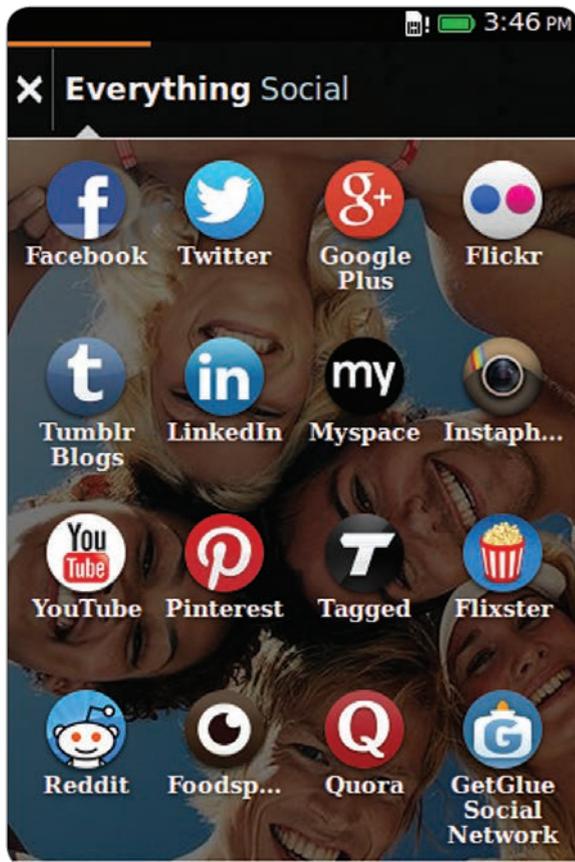
Даже если (что кажется вероятным) настольные ПК продолжит свое существование, вполне очевидно, что мобильные технологии продолжат расти и скоро составят львиную долю web-трафика. Многие считают *Firefox OS* всего лишь интересным побочным проектом от *Mozilla*, но по нашему мнению, ОС крайне важна для удержания доминантной позиции в войне браузеров.

С точки зрения технической перспективы, *Firefox OS* интересна потому, что все в пользовательском интерфейсе, от *Marketplace* и до набора номера, создано на web-технологиях, таких, как *HTML5* с *CSS* и *JavaScript*. Это означает, что все они вместе взятые обеспечивают ровную, бесшовную работу. Однако факт применения web-технологий не означает, что вам придется постоянно сидеть онлайн, чтобы извлечь из нее максимум пользы: система включает разные процедуры, позволяющие скачивать приложения для их использования оффлайн. С помощью фокусов типа *WebGL* разработчики также могут извлечь пользу из ускорения графики и прочих функций оборудования.

Интерфейс операционной системы не сверхреволюционен, но прост в использовании — частично, возможно, благодаря тому, что *Mozilla* изначально решила нацелиться на маломощные устройства в развивающихся странах. Более подробную информацию по первому потребительскому смартфону *Firefox OS* вы найдете в разделе обзоров (*ZTE Open*, стр. 16).



► Простой дизайн *Firefox OS* идеален для маломощных устройств.



Конкурирующие форматы приложений

Новая платформа смартфонов — будь она с открытым кодом или проприетарной — сталкивается с проблемой первичности курицы и яйца. Чтобы набрать пользователей, нужна платформа с кучей всяких приложений, а чтобы заставить разработчиков создавать приложения, нужны пользователи.

Появляющиеся мобильные операционные системы с открытым кодом предпринимают новый подход к этой проблеме. Ну, новый для мобильных устройств. Они облегчают создание приложений, работающих на разных платформах. Первый, и наиболее кросс-платформенный метод — *HTML5*. Хотя изначально он предназначался для создания web-приложений, эта экосистема стала достаточно мощной для создания многих основных приложений (*Firefox OS* ставит на то, что он достаточно мощный, чтобы вытеснить собой все родные телефонные приложения). Пока он только на начальной стадии, но судя по слухам среди разработчиков ОС, можно

ожидать, что формат пакетов будет один и тот же на большинстве или даже на всех операционных системах с открытым кодом. Это означает, что однажды разработанный код будет доступен везде, от телефонов *Android* до телефонов *ZTE* с *Firefox OS*.

Приложения *Android* работают на виртуальной машине *Dalvik*. Они могут также работать на *Alien Dalvik*, созданной *Muriad*. Однако пока что только *Jolla* подтвердила, что будет ее использовать, так что пользователям *Sailfish* придется устанавливать APKs, как и на устройствах *Android*. Рынок *Google Play* вряд ли будет доступен, но другие источники, вроде *F-Droid* и *Humble Bundles*, должны работать.

Amazon также сделал свой магазин приложений *Kindle Fire* доступным для скачивания (он тоже основан на *Dalvik* и формате *APK*). Так что в теории пользователи *Sailfish* будут иметь доступ к этой экосистеме с самого первого дня (хотя должны отметить, что до этого устройства мы пока не дотянулись).

Jolla, *Ubuntu Touch* и *Plasma Active* все построены на *Qt*, и *Tizen* также встроил этот инструментарий, хотя официально он не поддерживается. Это означает, что любая программа, разработанная на *Qt*, будет без проблем портироваться на эти три устройства, просто перепакетированием в соответствующий формат.

Конечно, суть экосистемы не просто в том, чтобы заставить программы работать на некоем устройстве. Суть в том, чтобы создать однородное взаимодействие с пользователем. Установка приложений *Android* на *Sailfish* будет примерно тем же, чем является установка приложений *GTK* на *KDE*: работать будут, но с ощущением чуждости.

Работа с кросс-платформенными приложениями даст телефонным системам хороший старт, но для обеспечения долгосрочного успеха необходимо привлечь разработчиков для создания родных приложений.

PLASMA ACTIVE

KDE для мобильных устройств.

Большинство мобильных устройств, от iPhone до Android, в первую очередь созданы для потребления контента. Они отлично подходят для того, чтобы смотреть видео, бродить по сети или слушать музыку. А вот для создания контента они уже не столь хороши. Беда тут невелика, ведь мало кто из нас тянется за мобильником, чтобы написать роман или слегка попрограммировать: для этого у нас есть компьютеры. Мобильное блаженство — устройство, способное объединить эти две области, то есть достаточно простое для работы с тачскрином и достаточно сложное, чтобы позволять делать настоящую работу.

Именно этого и пытаются добиться сообщество KDE вместе

«Plasma Active больше схожа с настольной ОС, чем любая другая.»

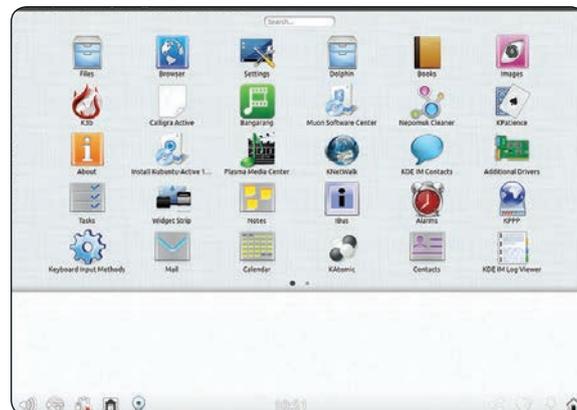
с BasysKom и open-six посредством Plasma Active. Это версия популярного рабочего стола KDE специально для мобильных

устройств. Поскольку он происходит из KDE, все популярные приложения KDE здесь доступны, в том числе и *Calligra Suite*.

Даже на беглый взгляд заметно наследие KDE, хотя в основном оно идет из темы, примененной в GUI, и набора значков. Здесь нет меню K для запуска приложений, и нет Window List. Вместо этого все происходит, когда вы тащите вниз верхнюю панель (наподобие получения доступа к уведомлениям в Android). Здесь вы найдете и список работающих приложений, и список установленных приложений, по которому можно осуществлять поиск. Что довольно странно для продукта от сообщества KDE — уж больно похоже на Dash из Gnome 3.

► Plasma Active больше всего похожа на настольную из всех мобильных ОС, которые мы тестировали, благодаря солидному списку функций.

Выпадающий список для отображения более сложных функций работает отлично, и, по нашему мнению, это лучший способ управления окнами из всех мобильных ОС, которыми мы пользовались. Но совершенно убойная функция Plasma Active — Activities. Так же, как и в KDE, это нечто наподобие небольших быстрых виртуальных рабочих столов; на них можно комбинировать виджеты и приложения для создания отдельных сред рабочего стола,



► Kubuntu Active показывает, что в мобильной гонке у Canonical есть не только Ubuntu Touch.

которые вы сможете листать. Чтобы включить Activities, надо просто перетащить вкладку с правой стороны экрана, затем выбрать необходимое из списка. Это движение выглядит вполне естественным в сенсорном интерфейсе.

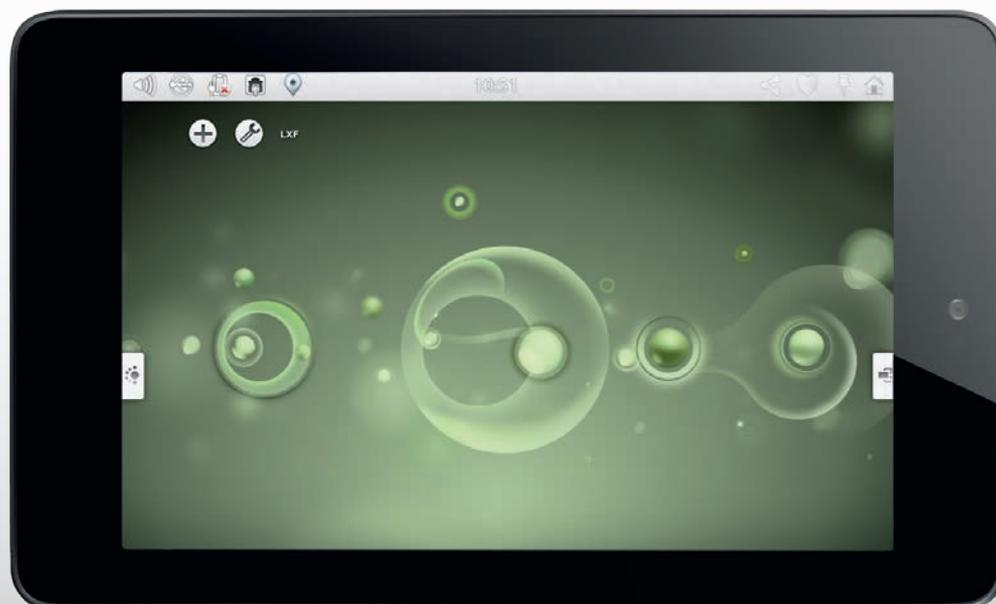
Хотя у Activities есть свои сторонники на настольных ПК, мы обнаружили, что мало кто из пользователей KDE считает их действительно удобными. Многих они приводят в замешательство, и нам еще предстоит увидеть дистрибутив с удобным набором Activities по умолчанию. Сможет ли мобильный мир стать местом их триумфа? Среди мобильных операционных систем, которые мы протестировали, Plasma Active была уникальна в том, что предлагала дистрибутивы, похожие на версии для настольного ПК и сервера. Задачей этих дистрибутивов будет создание правильного интерфейса — посмотрим, приложат ли они усилия для создания хорошего мобильного продукта.

Как и с Ubuntu Touch Scopes, подборка нужных activities для установки по умолчанию будет крайне важной задачей. Вряд ли многие пользователи захотят тратить время на настройку, добиваясь того, что они сочтут удобным и полезным. И подобно Ubuntu, это тоже та область, где KDE не преуспел на настольных ПК.

Plasma Active больше схожа с настольной ОС, чем любая другая из рассмотренных здесь. Можно ли считать ее конкурентом для Microsoft Surface Pro, опаснее, чем планшеты с Android и iOS? Нам кажется, что у нее наилучшие шансы перекинуть мост через пропасть, разделяющую устройства для потребления и для создания контента.

Возможность переключать activities в зависимости от текущего занятия станет истинным благом в этой области. Например, можно настроить одну activity для проверки электронной почты и работы в сети, а другую — для работы, как на настольном ПК с клавиатурой.

Возможно, из-за изобилия функций, Plasma Active, единственной из протестированных нами систем, создала ощущение, что наш Nexus 7 маловат. Немного дополнительного места на девяти- или десятидюймовом экране определенно не помешало бы, как и хороший чехол и клавиатура.



SAILFISH

Восстала из пепла Nokia.

Jolla, финская компания, созданная бывшими сотрудниками Nokia, строит новую мобильную операционную систему на базе технологий Qt поверх платформы Mer. Однако на этом сходство с системой Plasma Active заканчивается.

Sailfish (система Jolla) определенно является ОС для мобильных телефонов, без единой из сложных функций Plasma Active. Так же, как и у Ubuntu Touch, основной ее рыночный козырь — пользовательский интерфейс, и дизайнеры и инженеры в Jolla

заново обдумывают то, как нам взаимодействовать с нашими устройствами — как и Canonical.

Выбор функций управления у Jolla намного богаче, чем у Android и iOS. Однако две самых интересных области — это экран, который отображает работающие в данный момент приложения, и возможность создавать темы на основе изображений. Первая в той или иной степени присутствует на экране Home в большинстве других мобильных ОС, однако работает скорее как переключатель окон, разве что мощнее. Вторая позволяет изменять интерфейс, основываясь на цветах фонового изображения. Эта тема будет обволакивать все работающие приложения, а также экран Home и инструменты выбора приложений. Jolla делает ставку на то, что люди захотят изменять цветовые схемы своих телефонов, и первые устройства выйдут с заменяемыми корпусами, чтобы дать пользователям выбирать цвета.

К нам в руки пока что не попался ни один из этих телефонов, однако онлайн-видео и виртуальная машина Sailfish (часть SDK), показывают приятный интерфейс на Qt, использовать который, по нашему убеждению, будет сплошным удовольствием. Ожидается, что первые устройства появятся где-то в конце 2013 года.

«VM Sailfish (часть SDK) показывает приятный интерфейс на Qt.»



» Каждый прямоугольник соответствует работающему приложению, которое можно контролировать прямо с этого экрана Home.



Пробуем опции

Как отмечено в начале этой статьи, мы тестировали все на старом Nexus 7, выбранном не без причины: он не только поддерживается большим количеством ОС, чем любой другой портативный компьютер, но и обладает лучшими опциями для их загрузки.

В частности, *MultiROM*. Этот менеджер загрузки позволил нам легко установить ряд ОС на планшет и загружать разные ОС точно так же, как это можно делать с помощью *Grub* на ПК. У *MultiROM* нет сайта как такового. Вместо этого вся важная

информация (а также ссылки на ОС для установки) находятся на форуме, в ветке XDA Developers (<http://bit.ly/16EQdAf>). На момент написания статьи это работало только на старом Nexus 7, но разработчик трудится над созданием версии для современного планшета. Когда этот журнал окажется у вас на полке, версия, возможно, уже выйдет. Самые свежие новости см. на http://tasssadar.github.io/mrom_status.

Имитатор Firefox OS доступен в виде дополнения к *Firefox*. Он не даст вам полноценного ощущения

работы системы, потому что его не получится использовать как на телефоне, а мышь — не лучшая замена тачскрину; зато поможет оценить происходящее.

Подобным же образом, образ для виртуальной машины Tizen доступен на Tizen SDK, то есть на официальном сайте разработчика (<http://bit.ly/16o6mQj>), а live CD версии Plasma Active можно найти на сайте SUSE Studio (<http://bit.ly/14DcmkN>) и на сайте Ubuntu CD image (<http://bit.ly/16o6lq5>).

TIZEN

Из глубоких карманов Samsung и Co.

Большинство представленных здесь операционных систем разработаны или небольшими компаниями (как Sailfish от Jolla), или организациями — новичками на рынке мобильных технологий (как Firefox OS от Mozilla или Ubuntu Touch от Canonical). Tizen выделяется из общего ряда как яркая аномалия: ею занимаются такие колоссы, как Samsung и Intel, которые объединились, создав Tizen Association — под надзором Linux Foundation. Если бы деньги и пробивная сила решали на рынке мобильных ОС все, быть бы Tizen в дамках. Однако это не так, и за доказательством далеко ходить не надо: вспомним трудности Microsoft и Nokia, объединивших усилия в попытках продвижения Windows Phone на слабо заинтересованном рынке.

Интерфейс Tizen аккуратен и прост в использовании, но, честно говоря, трудно понять, что он предложит сверх готовых продуктов рынка... ну, помимо работы в недорогом секторе. Циничная часть нашей природы склонна предполагать, что основная

цель Tizen — помочь Samsung обрести более выгодную позицию, чем у Google, на переговорах об Android. Но наша роман-

«Трудно понять, что он предложит сверх готовых продуктов рынка.»

тичная сторона восхищена возможностями, которые предлагает Tizen в качестве мощного встраиваемого решения. Обзор платформы утверждает: «Tizen предназначен для того, чтобы предложить возможность полноценной унифицированной работы на нескольких устройствах — смартфоне, планшете, Smart TV,



Чистый интерфейс Tizen хорошо работает, но хватит ли у него примочек для привлечения пользователей к платформе?

ПК и автомобильной информационно-развлекательной системе — концентрируясь на мобильных платформах».

Мы не в силах изгнать подозрение, что успех платформы, если не сама ее идея, лежит в таких не избалованных сервисом (в отличие от мобильных) сферах, как ТВ и автомобильные информационно-развлекательные системы: платформа с открытым кодом может быть выгодна и потребителям, и производителям.

DESKTOP LINUX

Удобная альтернатива.

Пока что мы рассматривали ОС чисто для мобильных устройств, но почему бы не поставить на планшете обычный Linux (для нормального рабочего стола экран мобильного телефона слишком мал)? Одно из популярных решений —



Enlightenment, им мы и пользуемся. Помимо пары мелких придиорок, все работает хорошо. Все знакомые приложения работают, и все находится на знакомых местах. Однако быстро становится понятно, почему основные игроки возятся с системами для тачскринов. Иерархические меню, удобные в пользовательских интерфейсах с управлением мышью, при применении сенсорных технологий кажутся неестественными и сложными. Проблема в том, что сквозь собственный палец не видно, где в данный момент находится указатель. В таких интерфейсах, как правило, не работают приемы сенсорного экрана, непригодные для мыши (типа раздвижки пальцев для увеличения масштаба). Еще один виджет, который нельзя использовать и с мышью, и на сенсорном экране — полосы прокрутки. Использование обычного рабочего стола на планшете — неплохая идея, на практике вполне осуществимая, но все же хуже, чем ее альтернативы. Хотя Ubuntu продемонстрировал, что похожие пользовательские интерфейсы могут работать в разных форм-факторах, мы все же не уверены, что одной среде под силу нормально работать и на настольной системе, и на переносных устройствах. Экраны планшетов с высоким разрешением уже достаточно детально отображают рабочий стол, чтобы можно было работать с ними в этом размере. Но нам трудно представить переход на рабочий стол в основном интерфейсе планшета; это может быть удобный второй интерфейс для работы при наличии внешней клавиатуры и мыши. Хотя, возможно, как и Plasma Active, лучше оставить его более крупным планшетам.

Enlightenment хорошо работает на планшетах, но пользоваться интерфейсом без мыши бывает малоудобно.

ВЫЗОВ БРОШЕН

Как могут выиграть мобильные ОС Linux.

Готовясь писать данную статью, мы установили четыре из шести основных мобильных операционных систем Linux на 7-дюймовом планшете. Очередная ОС (Firefox OS) уже имеется на телефонах для продажи. Мы не смогли попробовать на реальном устройстве только Sailfish — ее выход вскоре планируется (возможно, уже в то время, как вы будете читать эту статью). Все вместе они представляют работу как минимум полдюжины компаний и множества преданных добровольцев.

Больше всего нас поразило разнообразие. Кроме CyanogenMod, ни одна из ОС не является клоном существующих решений: все они — подлинные инновации. Они предлагают пользователям возможность выбора способа работы мобильного устройства, и позволяют оптимизировать взаимодействие с пользователем согласно его потребностям.

Конечно, более десяти лет медленного роста использования рабочего стола Linux продемонстрировали нам, что одного этого недостаточно для достижения успеха. Сейчас два основных игрока на рынке — Google и Apple; это две крупнейшие компании на планете, и ни одна из них не будет в восторге от появления нового участника, особенно если новичок начнет преуспевать.

В некотором смысле мобильный рынок сегодня во многом похож на рынок настольных ПК в начале 2000-х — по крайней мере, в плане перспективы операционных систем с открытым кодом. Чтобы мобильные Linux ОС могли преуспеть, они должны учиться на ошибках настольного Linux. Нужно, чтобы они появились в магазинах, нужно отводить средства на рекламу, нужно, чтобы крутыми их считал среднестатистический парень с улицы, а не только технари. Короче говоря, им надо быть не просто лучше, чем существующие игроки; нужно, чтобы львиная доля потребителей считала их таковыми. Это суровое испытание, но есть ободряющие признаки. В промышленности курсируют слухи, что производители оборудования готовы поддержать нарождающиеся ОС, чтобы сломить дуополию Apple и Android. Носители рады предложить что-то новенькое, чтобы суметь выделиться. Прибавьте к этому тот факт, что средний пользователь смартфона обновляет свое устройство каждые два года, и вот вам быстро развивающаяся индустрия, созревшая для изменений. Как может свидетельствовать Blackberry, это рынок, способный меняться быстро и меняться сильно, но, как узнали на собственной шкуре Microsoft и Nokia, это также рынок, на который трудно пробиться. **LXF**



Гуру Google

Google многое сделал на благо открытого кода (уж точно не на зло).
Грэм Моррисон встретился с человеком, который практически
за всем ЭТИМ стоит.



Google только что отпраздновал свое пятнадцатилетие, и своим успехом во многом обязан Linux и открытому ПО. Это предоставило нам прекрасный повод узнать у менеджера открытых проектов Google, Криса ДиБона

[Chris DiBona], как открытый код влиял на Google — и наоборот. Увы, почему Android 4.4 окрестили Kit-Kat, прокомментировать он не смог.

LinuxFormat: За девять лет в Google, какие еще были важные сдвиги лично для вас, помимо запуска Android?

Крис ДиБона: Ну, когда я пришел в Google, там работало 1800 человек, а теперь уже более 44 000. При таком росте всего становится больше, верно? Приходят новые разработчики, им нужно больше исходного кода, появляются новые репозитории, — девять лет назад хлопот у нас было поменьше. А теперь это все на мне.

Когда я начинал, Android и Chrome еще не было, и непросто было сделать так, чтобы эти проекты развивались в соответствии с их собственными целями и целями Open Source. Взять даже только один Android. По нему существует больше 400 репозиториях Git, и нам пришлось создавать весь этот новый инструментарий, тоже с открытым кодом, в том числе Repo [инструмент управления репозиториями Android] и Gerrit [систему ревизии кода на основе web-технологий]. А затем и сам Git стал для нас неэффективным, поскольку не масштабировался до выпуска операционной системы. В конечном итоге, мы приняли на работу практически всю команду Git — пожалуй, найдется всего

пара человек, кто работает в Git и не работает в Google; так что Git существует на базе наших серверов и одновременно поддерживает наши клиенты в актуальном состоянии, а заодно работу системы в целом.

И теперь в Google есть, например, команда поддержки того, что вы называете Git в Debian. А это гарантирует, что когда пользователь Debian, Mac или Windows с помощью Git создает что-нибудь для Android или других проектов на его основе, он имеет дело с самой новой его версией. Тут довольно сложное хитросплетение.

LXF: В чем изначально виделась ваша роль в Google? Думали ли в компании: «У нас будет 100 открытых проектов, и нам нужен кто-то, чтобы ими управлять?»

«Учтите, открытые лицензии бывают довольно мудреными.»

КДБ: Будь их только 100, был бы другой разговор. По факту же, я полагаю, что всего выпустил более 3700 проектов — и крупных, и мелких; в основном, конечно, мелких. На каждый Android приходится по тысяче мелких проектов. Маленьких инструментов, которые кому-то пригодятся, и уйма заплаток-патчей.

Так что, нанимая меня, они знали только, что им нужен человек, который сумеет профессионально со всем управляться, поддерживать, так сказать, на ровном киле.

LXF: Как вы контролируете соответствие стандартам Open Source в таких проектах, как Android?

КДБ: В Android главный не я, но я им помогаю. С Android нам повезло в том, что соответствие с самого начала удалось сделать частью самого инструментария и системы сборки.

LXF: За годы до их официального выхода?

КДБ: Да, года за три. Мы сотрудничали с командой Android и, на самом деле, создали инфраструктуру команды Android в мировом масштабе, вместе с партнерами Android, и прочее. Мы можем не просто сказать им: «Эй! Вам бы надо соответствовать стандартам!» Потому что этого мало. Мы можем сказать: «Вот так вы будете соответствовать». И теперь, фактически, когда вы имеете дело с любым устройством на Android, даже не пользуясь при этом Google

и не желая ни во что вникать, — это следствие совместимости Android, ведь именно там будут составляться все разделы «Справки», как и все прочее, чего требуют открытые лицензии. Так что

встретить не соответствующее стандартам устройство Android — я имею в виду буквальное соответствие — можно очень редко. Даже при полном незнании требований открытых лицензий. Иногда это самый лучший вариант, верно?

Даже в компании вроде Google, где все на местах, когда создаются новые репозитории, стартуют новые проекты — все не так гладко. Мы должны быть полностью в курсе того, что происходит в компании, чтобы на этапе запуска нового проекта гарантировать его соответствие



открытым лицензиям и, собственно, использование самых актуальных версий свободного ПО — в плане исправления ошибок и прочего. Мы стараемся подключиться как можно раньше, чтобы не препятствовать старту проекта, ведь если мы замедляем пуск, мы подводим Google в целом, нам бы не хотелось быть той самой группой.

LXF: Выходит, что соответствие у сотрудников Google уже в крови?

КДБ: И да, и нет. Учтите, открытые лицензии бывают довольно мудреными. И вовсе не обязательно заставлять инженеров в это вникать, ведь если голова у них занята нюансами взаимодействий, разработка идет гораздо медленнее. Мы стараемся давать им общие рекомендации и умные инструменты, чтобы они понимали то, что подразумевается в их проекте. Мы не хотим делать из них экспертов по лицензиям, поскольку это тоже своего рода баланс: преуспевая в одном, можно ли добиться того же в другом? Возможно, но по-моему,

лучше дать им полностью сконцентрироваться на разработке и прочем.

LXF: Изменился ли подход Google к открытому коду за эти 9 лет?

КДБ: Конечно. Забавно, но в каждом проекте он особый.

«В ChromeOS иной подход к разработке ОС, нежели в Android.»

LXF: То есть в ChromeOS подход иной, нежели в Android?

КДБ: Можно сказать, что да. В том смысле, что в ChromeOS иной подход к разработке операционной системы, нежели в Android. И это довольно хитрая штука, потому как создавая, к примеру, браузер, вы захотите сделать некоторые плагины

более надежными, но эти плагины являются по умолчанию закрытыми — тот же Flash.

И если вы хотите, чтобы ChromeOS на вашей машине воспроизводила содержимое Flash, придется заключить специальное соглашение с Android, позволяющее использовать данную версию Flash таким образом. И ведь в *Chromium* или *ChromiumOS* такого не было, верно? Это появилось только в ChromeOS.

Вот такие хитросплетения есть между открытым и проприетарным ПО, и как же с этим быть?

Аналогично, когда вы желаете договориться с любым производителем контента из США, Европы и остального мира, они заинтересованы в том, чтобы получить полностью «защищенный канал» и гарантии, что этот договор не будет нарушать требования открытых лицензий в рамках данной операционной системы или программы. Здесь может быть куча подвохов.

Помимо всего этого, есть еще работа над стандартами. Скажем, одна часть компании, занимающаяся зашифрованными медиа-расширениями, разрешает использовать плееры Netflix и тому подобное, а затем Ян Хиксон [Jan Hickson], который работает на меня, говорит, что это не входит в Web WG (Web Hypertext Application Technology Working Group) спецификацию для HTML 5.

LXF: И кто принимает решение?

КДБ: В этом, пожалуй, особенность Google. Мы используем и то, и другое. Или ни то, ни другое, в зависимости от целей. Иногда при этом появляются некоторые вещи, которые кажутся конфликтными, но лишь на первый взгляд. Для нас вполне допустимо рассматривать спецификацию Web WG HTML 5 просто как документ, не зависящий от каких-либо запатентованных технологий, как это обстоит в некоторых других спецификациях, используя при этом дополнительную спецификацию HTML 5, разрешающую ее использовать. Мозг сломашь.

LXF: Вы имеете право голоса при принятии таких решений, или сохраняете нейтралитет?

КДБ: Я руководствуюсь только одним принципом, и до сих пор он меня не подводил. И он таков: неважно, нравится ли мне проект, каковы его возможности, являюсь ли я его пользователем, но пока он соответствует открытым лицензиям — по духу и по букве — я доволен. Мои личные пристрастия при этом значения не имеют — пока к моим друзьям из открытого ПО относятся уважительно. Пока с ними обращаются по достоинству, мое личное отношение здесь ни при чем, потому как я сам люблю, когда все по правилам. Я — своего рода хранитель этих правил, и пока нарушений нет, я доволен.

LXF: И это включает такие спорные территории, как DRM в спецификации HTML 5?

КДБ: Там все крайне спорно. Я пришел к тому, что в большинстве случаев просто стараюсь, насколько это в моих силах, поддерживать нормальные взаимоотношения между двумя командами, без каких-либо нападок и без попыток продавить решение проблемы в ущерб другой стороне.

LXF: И пока они в согласии, вы довольны?

КДБ: Пока и те, и другие в согласии, мы все довольны. На самом деле, с этим никогда особо не было проблем. Всем известно, что для нас это важно, так что проблем не возникало. Сложно, когда люди работают в конфликтных или конкурентных проектах: очень сложно избежать перехода на личности. Как управленцу, занимающийся разработкой, мне это известно.

LXF: А если имеет место различие философий?

КДБ: Забавно, что люди говорят: «Это же просто программа, не принимай близко к сердцу». Или «Это же просто бизнес, ничего личного». Но забывают, что на самом деле и то, и другое — тоже личное. Это то, чем мы занимаемся по жизни, и это важная ее часть, которую нельзя упускать из вида. Тут вы можете сказать «Так и ты продался, Крис?» Но у меня нет такого ощущения, поскольку в контексте деятельности всей компании, достаточно активной и благонамеренной, я считаю, что мы должны об этом говорить, особенно внутри.

LXF: Чему вы уделяете больше всего времени сейчас?

КДБ: Сейчас у меня есть команда, человек 30, и работаем мы по разным направлениям соответствия стандартам и над Summer of Code, также занимаемся созданием инструментария и инфраструктуры. Лично я, по большей части, руковожу людьми. Довольно много времени уходит на подбор кадров.

При таком количестве инженеров, руководителей программ, руководителей проектов, как у нас, все они должны думать о своей



О КОНФЛИКТАХ ИНТЕРЕСОВ

«Пока к моим друзьям из открытого ПО относятся уважительно, мое личное отношение ни при чем.»

карьере и работать в Google с удовольствием. Так что продвижение и подбор кадров, на мой взгляд, очень важны, но вашим читателям это, должно быть, не особо интересно.

LXF: В политику приходится вмешиваться?

КДБ: И да, и нет. То есть, в компании я один из директоров. То есть по должности мне не положено. Ларри — положено, и есть еще ряд людей, гораздо более влиятельных. Но я могу им во многом быть полезен, могу помочь людям сделать карьеру, и это себя действительно оправдывает.

LXF: В чем, по-вашему, состоит самый большой вклад Google в развитие открытого ПО?

КДБ: Думаю, три самых главных достижения — это Android, *Chromium* и Go. Если бы лет через 10 вы бы спросили меня, чем я горжусь больше, я бы затруднился с ответом, потому что Android имеет огромный резонанс.

LXF: Есть еще Summer of Code!

КДБ: Ну, Summer of Code — вещь сугубо личная, изменившая судьбы тысяч людей. А с Android и *Chromium* счет идет на миллионы и миллиарды. Но своим сердцем компьютерщика, своей душой программиста я обращен к таким проектам, как Summer of Code, как Go и даже GCI, нашей программе для высшей школы. И понимаю, что в некотором роде такие вещи укрепляют позиции открытого ПО даже сильнее, чем Android и *Chromium*. Последние, на данный момент, — это рыночные силы.

LXF: Они доказывают, что открытое ПО имеет право на существование?

КДБ: Они фундаментальны. Им не нужно ничего доказывать. За последние лет 20 всегда находились люди, считающие открытое ПО злокачественной опухолью, или незаконным, или врагом. Они такое говорят, а я не могу понять, зачем, ведь открытое ПО повсюду. И если нас это устраивает, и мы хотим, чтобы компьютерная наука развивалась, нам нужно продолжать его укреплять. Этим мы и занимаемся, при помощи языков и Android. Совершенствуя существующие открытые платформы, такие как Сеть, посредством *Chromium*. Нужно понимать, что до появления *Chrome* люди использовали браузеры, где одна вкладка могли обрушить всю ОС. Сейчас это практически невозможно. Это кажется нелепым. И там, где о производительности Java никто не заботился, с *Chrome* мы смогли изменить ситуацию так, что все увидели в этом смысл. А также заставить тех, кто работает на WebKit и Gecko, пересмотреть свои подходы к производительности и безопасности.

Без всего этого, дела в Сети сегодня обсто- яли бы гораздо хуже, ведь там полно вредоносного ПО, мошенников, желающих вас обдурить. Но благодаря *Chromium*, мы достаточно быстро стали искать решения, что, по-моему, во многом спасло нас от того, во что мог превратиться Интернет. Можете ли вы представить, что у вас нет защиты от угроз или обособления процессов, которые из *Chrome* перешли и в другие браузеры?



» ДиБона: «Мне виделось, что Summer of Code поможет привлечь новых людей в открытое ПО».

Можно ли сейчас просто так бродить по Интернету? Картинка довольно мрачная. Слишком много вредоносных программ. В конце концов, придет к ограничению посещаемых сайтов, ограничению вариантов просмотра и так далее.

LXF: Думаете, Google выжил бы без открытого ПО или Linux?

КДБ: Возможно. Но Интернет — навряд ли. Так что Google могло и вовсе не быть. Возможно, он был бы чем-то другим; но без открытого ПО, питающего Интернет, у Google было бы куда меньше Интернет-пространства для освоения, для размещения рекламы, меньше идей для развития Android и *Chromium*. Я думаю, они неотделимы друг от друга.

LXF: Из 1200 участников Summer of Code из 60 стран, 271 были/остаются жителями Индии. Думаете, в последующие 10 лет изменятся центры технической мысли и ее направления?

КДБ: Надеюсь. Каждый год мы наблюдаем увеличение количества участников не из США. До-

есть пара университетов, которые держатся за Summer of Code как часть своей образовательной программы. Только вдумайтесь! 79 студентов, изучающих информатику, из маленького университета в маленькой стране, в разгар гражданской войны, делают очень достойные вещи. Вот где надежда Интернета и будущего компьютерных наук вместе взятых.

LXF: Таким вы себе и представляли Summer of Code?

КДБ: Не совсем. Не хочется строить из себя провидца. Такого я никогда представить не мог. Мне виделось, что Summer of Code поможет привлечь новых людей в открытое ПО. Тех, кого раньше мы бы не увидели никогда — потому что нам приходилось буквально спонсировать участников, чтобы им не пришлось летом разъезжаться по домам и заниматься чем угодно, только не информатикой. Так что для меня это был просто способ чем-то занять юных исследователей на лето, и желательным, связанным с открытым ПО. Но вылилось все в нечто большее. Превзошло все мои ожидания,

и оправдало усилия всех тех команд, которые приезжали, давали уроки и прочее. Вспомните, у каждого разработчика в Summer of Code есть свой наставник и проект. Без наставников ничего бы не вышло.

Главным прорывом Summer of Code является то... что он сводит вместе опытного разработчика, привыкшего работать удаленно, и новичка. И это потрясающе, когда студент может лично обратиться к своему наставнику и сказать: «У меня такая-то проблема». Или наставник может посмотреть лист изменений и сказать: «У тебя проблема в том-то. Если ты сделаешь так и так, все будет хорошо. Если сделаешь то-то и то-то, будет плохо». Такого даже на работе практически не бывает. **LXF**

«Люди забывают о том, что программы и бизнес — тоже личное.»

ля США по-прежнему остается существенной — 250 или приблизительно так — но откуда только ни приезжают; из той же Шри Ланка. Даже пока там шла гражданская война, тамилы и другие тамошние студенты приезжали на Summer of Code; и диву даешься, как это событие дошло даже до такой страны?

Так что Шри Ланка нам, в некоторых отношениях, пока даже интереснее, чем Индия и Китай. Вот вам очень малая нация, и если приглядеться,



Поколение: Код

Лес Паундер исследует инициативы, которые призваны научить новое поколение программированию.

В сентябре британское правительство сообщило, что начиная с 2014 года, в учебный план по информационно-коммуникационным технологиям (ИКТ) будет внесено несколько изменений. В течение целого поколения детей учили работать с компьютером в программах *Microsoft Office* и создавать документы, электронные таблицы и другие офисные документы, например, презентации *PowerPoint*. Но начиная с 2014 года, школьников вместо ИКТ

будут обучать компьютерным наукам [*Computer Science*]. Это означает, что за оставшееся время учителя должны разработать собственные временные курсы.

Также было признано, что если у учителей средней школы есть доступ к множеству ресурсов, то учителя начальной школы обеспечены не так хорошо, и стало очевидным, что в начальных школах нужно не только учить детей программированию с раннего возраста, но и обучить

персонал и укомплектовать школы необходимым оборудованием.

К счастью, возникло несколько групп энтузиастов, работающих над созданием общественного подхода к проблеме, и это привело к увеличению количества детей, занимающихся программированием, что помогает преодолевать основную тенденцию.

Наряду с группами энтузиастов есть несколько сайтов и курсов, предназначенных для того, чтобы

дети познакомились с программированием; наиболее доступные для детей — www.codecademy.com и www.codeavengers.com. Эти сайты помогают детям (и взрослым) изучать языки программирования спокойно и постепенно и позволяют создать прекрасную основу для урока.

Существует много объединений, которые хотят изменить способ преподавания программирования в школах, и в этом номере мы познакомимся с двумя из них: Young Rewired State, ежегодная конференция для детей по всему миру, и Code Club, еженедельное собрание, проводимое среди британских школ во внеклассное время. Мы спросили команды каждой группы о том, как они были созданы, что движет их вперед и об их наибольших успехах на данный момент.

Young Rewired State

Young Rewired State — мероприятие продолжительностью в неделю, когда дети в центрах по всей Великобритании работают вместе как команда разработчиков над созданием проекта на основе открытых данных. Команда изучает основы управления временем и проектами и приучается писать код к конкретному сроку.

У команд есть наставники, предоставляющие поддержку и руководство, но не выполняющие работу за команду. Кульминация деятельности команды — двухдневное событие в Бирмингеме, на котором до 500 детей представляют свои проекты серии судей, и каждая команда надеется пройти в финал. Цель Young Rewired State — поощрить детей работать в команде и изучить основы программирования для последующего трудоустройства.

Недавно нам удалось поговорить с Томом Бруксом [Thom Brooks], который работает специалистом по связям с общественностью и коммуникациям Rewired State.

Linux Format: Привет, Том! Не расскажете, что такое Rewired State и каковы ее цели?

Том Брукс: Rewired State и Young Rewired State основаны Эммой МалКуини, известной в Твиттере под ником @hubmum. Rewired State разрабатывает и организует мероприятия, позволяющие объединить креативных разработчиков и отраслевых экспертов для решения реальных проблем.

Наши клиенты включают правительства, промышленных гигантов, конференции и благотворительные организации. Rewired State дала жизнь нашей первой конференции в марте 2009 года, когда более сотни разработчиков поработали вместе над печально известным Национальным днем «Взлома правительства [National Hack the Government Day]». Событие стало катализатором для знаменитого сайта <http://data.gov.uk> и широко признается источником вдохновения, лежащего за усилиями правительства Великобритании сделать данные открытыми.



LXF: Как начался проект Young Rewired State и как прошли его первые дни?

ТБ: В 2009 году мы решили организовать мероприятие выходного дня под названием Young Rewired State, на котором сотрудники лондонских офисов Google представляли

► YRS проводит ежегодный Фестиваль кода для тех, кому нет восемнадцати.



открытые правительственные данные юным программистам Великобритании. Предвкушая встречу с юными программистами, мы открыли двери для 50 участников (ограничение из-за вместимости лондонских офисов Google)... а зарегистрировались [всего] трое. Обзвонив школы и встряхнув Интернет, мы поняли, что проблема гораздо глубже, чем незаинтересованность детей открытыми правительственными данными. В стране не хватает молодых программистов, и есть только отдельные дети, которые учатся программировать, сидя по своим спальням, а их родителей ужасают мысли, что из отпрысков не выйдет ничего путного. Школы часто выявляют учеников-одиночек, которым это интересно, но помочь им нечем, так как программирование давно перестали преподавать.

Затем мы потратили три месяца на поиск 50 кандидатов, и с огромным облегчением и еще большим

предвкушением свели их вместе. Мы провели с ними выходные, с наставниками и экспертами по правительственным данным под рукой, и смотрели, как они работали вместе и создавали приложения и сайты, все на основе открытых правительственных данных.

На наших глазах создавалось столь необходимое сообщество — никогда больше эти юные гении не будут программировать в одиночку, с этого момента у них есть свои соратники и наставники, которые будут управлять их обучением и превращением в увлеченных взрослых программистов.

LXF: В чем цель Young Rewired State?

ТБ: Цель — создать всемирную независимую сеть молодых программистов с наставниками, поддерживаемую и поддерживающую самостоятельное обучение и в конечном счете решение реальных проблем.

LXF: Young Rewired State широко освещалась в СМИ последние несколько лет, но можете ли вы сказать, как она выросла за это время?

ТБ: В 2009 году мы начали с 50 ребят, а сейчас, в 2013-м, их больше тысячи. В этом году мы даже вышли за пределы Великобритании, проведя свое первое «международное» мероприятие в Нью-Йорке ранее в этом году.

Наши планы по расширению также включают мероприятия в Европе — в Берлине в сентябре, а также в Йоханнесбурге и Сан-Франциско в этом году. Мы планируем следовать той же модели: найти основу в 50 человек и наращивать ее. Наше первое международное событие (так мы называем наше расширение

► Кульминация Фестиваля кода — двухдневная презентация на Custard Factory в Бирмингеме.

«Предвкушая встречу, мы открыли двери... [всего] трое.»

за пределы Великобритании) было очень успешным, и 50 ребят мы нашли с легкостью!

В этом году на Фестивале кода мы собрали более 500 человек на Custard Factory в Бирмингеме.

Заведение трещало по швам, и нам, похоже, в следующем году придется искать новое помещение с большей вместимостью. На Фестивале кода в этом году мы сделали более 125 проектов! В прошлом у нас было 100. Кажется, мы растем и хорошеем. Это было по-настоящему масштабно и удивительно!

LXF: Количество центров Young Rewired росло с первого дня, но не расскажете, как вы работали с другими, чтобы построить несколько центров?

ТБ: Сообщество очень важно для нас. Именно благодаря ему Young Rewired State выделяется на общем фоне. Центры и наставники играют огромную роль в успехе Young Rewired State.

В этом году у нас было 40 специализированных центров по всей стране, от Абердина до Портсмута. Наше сообщество по-настоящему хорошо своим желанием отдачи. Центры понимают необходимость воспитывать молодые таланты, и мы тесно работаем с ними, чтобы неделя была успешной. Мы предоставляем свой шаблон того, как сделать неделю наилучшей, и назначаем наставников, которые ассистируют и помогают. В этом году у нас было более 160 наставников, и это удивительно. Думаю, что чем больше людей узнают о необходимости этих базовых знаний, тем больше людей начнут спрашивать, как к нам присоединиться. Лучший способ — организовать центр. В этом году организовать нашу замечательную неделю нам помогли независимые предприятия малого бизнеса, известные крупные медиа-центры, школы и хостинг-центры.

► В число музыкантов на Фестивале кода 2013 вошел Герой Термена [Theremin Hero] на лазерной арфе.

LXF: Young Rewired State получила огромную поддержку, но какой момент был самым выдающимся?

ТБ: Всего один? Я думаю, выдающийся момент — просто видеть, как все это работает. Этот процесс такой органичный и дает такую

большую отдачу. Ну, почти органичный, большая часть работы происходит за кулисами в очень маленькой команде! Ажиотаж в Твиттере... результат был удивительным. Думаю, выдающийся момент наступил вечером в воскресенье, когда все огляделись и подумали: «Ух ты, я участвовал в чем-то удивительном. Жду не дожусь следующего года».

LXF: Чего вы надеетесь достичь с Young Rewired State в следующие 12 месяцев?



► Клер Сатклифф, сооснователь Code Club: «У нас есть 990 клубов по всей Великобритании».

ТБ: Нас ждут Берлин и другие международные события. На 2014 год уже присматриваемся к Нью-Йорку. Мы планируем следовать той модели, с которой начали, с небольшим изменением. Надеемся в 2014-м организовать в Нью-Йорке пять центров. Мы также хотим, чтобы сообщество продолжало расти, воспитывая все больше увлекающихся программированием ребят и открывая для них открытые данные и сообщество и просто получая удовольствие от всего этого.

Если вам кажется, что этот год был успешным, ждем вас в 2014-м! Если вы хотите поработать с Young Rewired State, зайдите на наш сайт <https://youngrewiredstate.org> и узнайте о центрах и группах в том месте, где вы живете.

Code Club

Не похожий на Young Rewired State, но не менее влиятельный подход — Code Club. Это еженедельный факультатив, тематически разделенный по четвертям, во время которого дети учатся программировать по учебному плану, разработанному командой и сетью волонтеров. Большинство волонтеров приходят из сообщества, они разрабатывают для Code Club уроки по Scratch и web-технологиям и привносят свой уровень знаний в классную комнату.

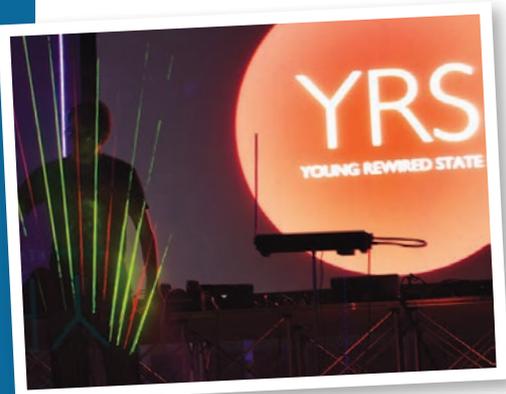
Недавно я поговорил с Клер Сатклифф [Clare Sutcliffe], которая совместно с Линдой Сэндвик [Linda Sandvik] основала Code Club в апреле 2012 года. Клер рассказала нам подробнее о проекте и о том, как заинтересовавшиеся могут присоединиться.

Linux Format: Привет, Клер! Прежде всего, не расскажете ли нам о команде, работающей над Code Club?

Клер Сатклифф: Code Club основали Линда Сэндвик, дизайнер интерфейсов, и я, Клер Сатклифф, графический дизайнер. Но двигаться дальше проекту помогают волонтеры, которые тратят свое время на его поддержку.

LXF: Клер, можете сказать, как появилась идея создания Code Club?

КС: Линда и я сидели в пабе и после нескольких бокалов заговорили о том, чтобы провести день программирования, который должен помочь детям научиться программировать, так как в то время их не учили программированию по учебному плану ИКТ в школе. Затем мы решили сделать такие встречи регулярными. Это познакомило меня с массой программистов, которые захотели помочь.



Стать посланником STEM

Посланники STEM (Science, Technology, Engineering and Mathematics — наука, технология, инженерное искусство и математика) — волонтеры, которые рассказывают об этих предметах школьникам, чтобы те получили новые знания для будущего трудоустройства и увеличили количество талантов в Великобритании.

Обычно посланники работают в соответствующих областях, и все их резюме

исследуются STEMNET, ведущей организацией, которая выполняет полагающиеся обязательные проверки.

После принятия в ассоциацию посланники регулярно получают информацию от региональных представителей: где предоставляется возможность поработать со школами в его регионе. За подробной информацией загляните на сайт www.stemnet.org.uk.

Мы окончательно запустили Code Club в сентябре 2012-го года, и через двенадцать месяцев у нас есть 990 клубов по всей Великобритании.

LXF: В целом, кто такие волонтеры Code Club и какие требования к ним предъявляются?

КС: Они совсем разные, но их ядро, около 45% — это учителя в школах, которые проводят Code Club. Остальные 55% — фрилансеры, родители и программисты на пенсии.

Волонтеры Code Club должны пройти проверку DBS (Disclosure and Barring Service — Служба по раскрытию информации и лишению права занимать определенные должности), ранее известную как проверку CRB (Criminal Records Bureau — Бюро криминальной истории). Code Club рекомендует волонтерам также зарегистрироваться как посланники STEM (подробности см. во врезке «Стать посланником STEM», стр. 44). Посланники работают со школами для продвижения всех предметов STEM и поддерживают учителей по этим предметам практически. Стать посланником STEM — несомненно, отличная возможность поделиться своей любовью к науке, технологии, инженерным искусствам и математике, поддерживая детей, и в ответ на заявку STEM все равно выполняет проверку DBS. Подробности можно найти на нашем сайте <https://www.codeclub.org.uk/start-a-club/volunteers>.

LXF: Что происходит на типичном занятии Code Club?

КС: Занятия продолжительностью до часа проходят раз в неделю и тематически разделены по четвертям. Каждую четверть мы работаем над новой темой: например, в первую четверть — над Scratch, во вторую — над продвинутым Scratch, где по-настоящему расширяем знания группы в Scratch. В третьей четверти мы изучаем HTML и CSS и рассказываем группе о строительных блоках Всемирной паутины. Подробная информация есть на нашем сайте (<https://www.codeclub.org.uk/start-a-club/volunteers>) в разделе «Term Outlines [Учебные планы на четверть]».

Мы просим, чтобы все волонтеры к каждому уроку обязательно готовили примеры кода, распечатывали раздаточные материалы и были знакомы с примерами и их назначением. Каждый проект Code Club создается с конкретной темой, которую легко раскрыть за час при минимальном участии волонтера. Но мы понимаем, что все клубы разные, и они могут свободно менять эту схему по своему усмотрению.

LXF: Что было вашим главным успехом?

КС: Думаю, наш главный успех — это наш рост за последние 12 месяцев: сейчас у нас 990 центров в Великобритании, и благодаря нашей фантастической сети волонтеров появляются новые центры.

LXF: Насколько сложно школьному учителю и волонтеру поработать со школой, чтобы создать центр Code Club?

КС: Учителю достаточно зайти на наш сайт и зарегистрировать свою школу — естественно, согласовав это с руководством.

Для волонтера это несколько сложнее, так как им нужно прийти в местную школу и добиться одобрения директора школы. Затем школе нужно зарегистрироваться на сайте. После



Обучающие ресурсы

» **Scratch** Пожалуй, самый популярный инструмент обучения программированию для детей от шести лет и старше. В нем для построения программ используются блоки с инструкциями.

<http://scratch.mit.edu>

» **Hackasaurus** Одна из простых утилит JavaScript от Mozilla. Она есть в ваших закладках и позволяет редактировать другие сайты: зайдите на сайт, щелкните на закладке и принимайтесь за редактирование. После обновления сайт возвращается к исходному виду.

<http://hackasaurus.org/en-US>

» **Code Academy** Бесплатная система интернет-обучения с курсами по JavaScript, PHP, Python и Ruby. Уроки изложены сжато и имеют равномерную кривую обучения.

www.codecademy.com/learn

» **Blender** Не самый очевидный вариант, но *Blender* очень привлекателен для ребят. Они могут много узнать о математике, физике и разработке игр. В *Blender* также имеется собственный интерпретатор Python, который позволяет одновременно изучать Python.

www.blender.org

регистрации волонтер и школа начинают работать вместе для выполнения дальнейших действий. Те, кому интересно подробнее узнать о Code Club, могут зайти на сайт www.codeclub.org.uk.

Будущее

С изменениями в учебных программах, прогрессивном мышлении групп, о которых мы рассказали в этой статье, и, конечно же, вдохновляющем, безудержном любопытстве детей будущее выглядит довольно оптимистичным. Волонтеры в Великобритании работают со школами, колледжами и другими организациями, такими как объединения скаутов [Scouts and Guides], предоставляя им доступ к знаниям, времени и оборудованию, что позволяет нашим детям получить то, чего не хватало предыдущему поколению: знания для того, чтобы создавать, делиться и изобретать новое.

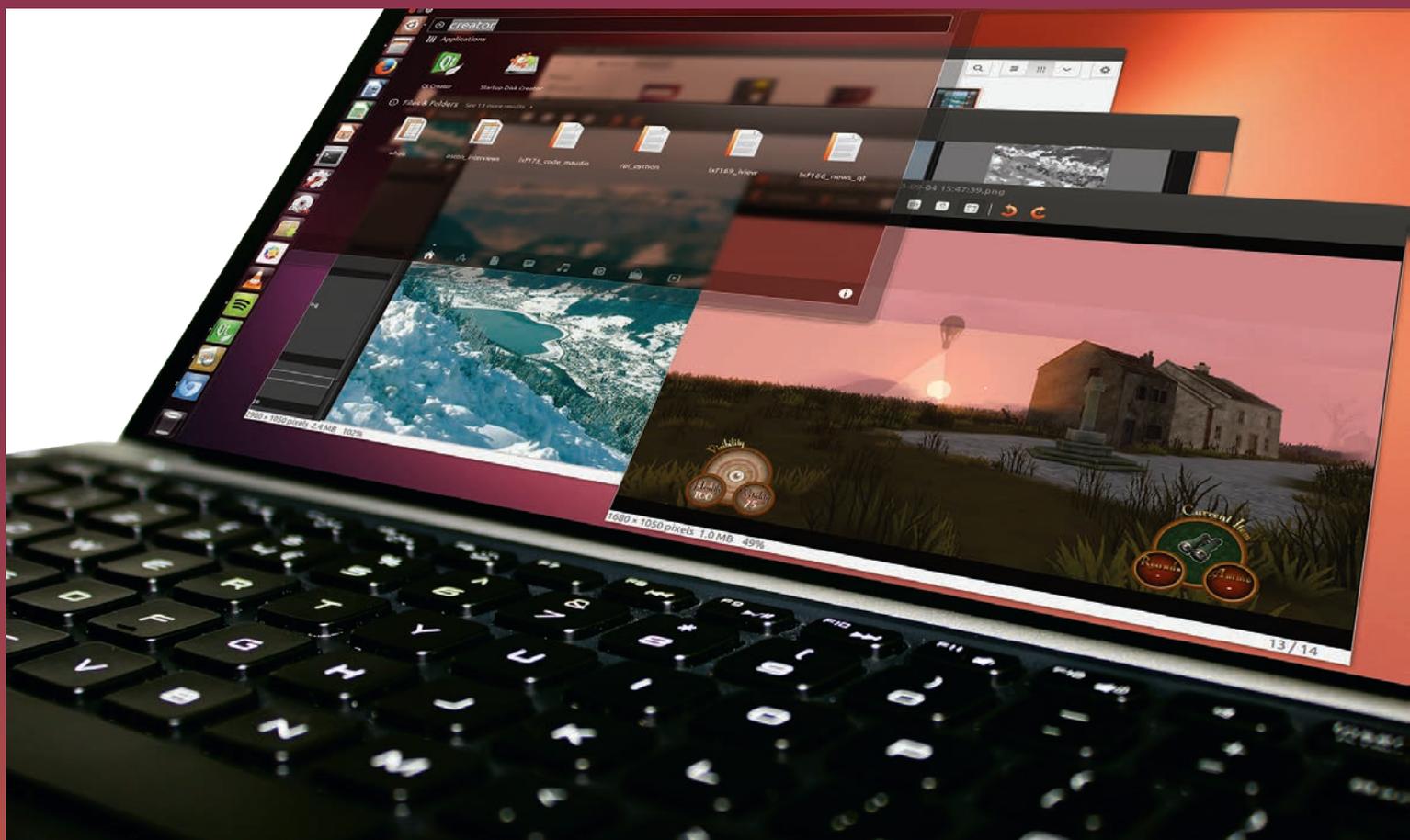
Сейчас учителя, получившие новую квалификацию, приносят с собой идеи для уроков и пользуются новыми технологиями, чтобы привлечь внимание класса. Мы знаем учителей, разрабатывающих новые способы обучения в классе с помощью Arduino и Raspberry Pi; один из ярчайших примеров — роботы Саймона Уолтера [Simon Walter], находящие выход из лабиринта и работающие под управлением Scratch и Raspberry Pi.

Среды программирования, такие как Scratch, и средства веб-разработки, такие как Thimble и Hackasaurus от Mozilla, помогают детям увлечься кодом, изучить, как он работает и что можно сделать с его помощью.

Мы не можем не упомянуть Raspberry Pi, прекрасное средство обучения для школы и дома во всем мире. Мы видим, как это движение развивается, и с ростом культуры созидания в Великобритании у создания, разделения и развития технологий появляется новая грань. Вопрос в том, скоро ли это коснется уроков труда? Мы очень хотим увидеть на уроках труда станки для лазерной резки, трехмерную печать и Arduino. Культура созидания, которая включает электронику, программирования и основные знания с уроков труда — это следующий шаг, и мы надеемся, что школы придут к этому как можно раньше. **LXF**



«Учителя, получившие новую квалификацию, вносят идеи уроков.»



ПРОЕКТ SPUTNIK ИЗНУТРИ: DELL UBUNTU ULTRABOOK

Грэм Моррисон тестирует Dell XPS 13 Developer Edition и беседует с теми, кто стоит за этим проектом.



► Barton Джордж, основатель и руководитель Project Sputnik.

В середине прошлого года до нас начали доходить слухи о том, что Dell подумывает о создании Linux Ultrabook в сотрудничестве с Canonical. Это был Project Sputnik. В Dell инициатором проекта был Бартон Джордж [Barton George], директор программ развития облачных технологий Dell [Dell's Director of Cloud Developer Programs], и мы смогли побеседовать с ним о том, как начинался Sputnik, чего ожидать от инициативы и как в Dell смотрят на оборудование Linux. Кроме того, компания была настолько любезна, что предоставила нам одно из своих новых устройств, и последние несколько месяцев мы активно использовали его в Linux Format. Оно сопровождало нас в Портленде на конференции OSCON, и мы

употребляли его для всех наших учебников: от кодирования и работы в облаке до работы с аудио и игр. Мы таскали его с одного заседания на другое, давали его истощенным батареям отчаянную 20-минутную зарядку от незаконной розетки в Chicago O'Hare и искромсали оригинал Ubuntu, чтобы извлечь максимум из оборудования.

Ultrabook — это миниатюрный форм-фактор для ноутбуков, сочетающий портативность нетбука с производительностью крупной машины. Они идеальны для разработчиков, желающих вкалывать и во время поездок, а не просто проверять почту. Имея достойный CPU и солидный объем памяти, ультрабук позволяет, например, компилировать большие деревья исходников, работать

с виртуальными машинами и играть в требовательные к ресурсам игры; все это было бы невозможно на нетбуке, маломощном ноутбуке или планшете.

Project Sputnik является именно такой разновидностью ноутбука, прямиком нацеленной на разработчиков; но в нем есть все, чтобы привлечь любого поклонника Linux, жаждущего союза портативности и мощности. Да, вы можете установить Linux практически на любом ПК. Но для большинства из нас это необходимость, а не выбор. С куда большим удовольствием мы купили бы оборудование от солидного производителя, с должной поддержкой Linux, и нам не пришлось бы объяснять, почему мы не нуждаемся в проверке на вирусы при покупке. Особенно если наша работа зависит от оборудования. Вот почему так важен Dell Project Sputnik. Это прекрасно разработанный ультрабук от третьего по величине производителя ПК, и он поставляется с Ubuntu.

Бартон рассказал нам о своих причинах затеять проект: «Изначально идею выдвинул Стивен О'Грейди [Stephen O'Grady] из RedMonk (известной аналитической компании с нестандартным подходом к технологии), — сказал он. — Мы-то уделяли все внимание Web и тому, как лучше понять ситуацию и продавать свою продукцию всем этим Google и Facebook'ам, доводя ее до стартапов и студенческих комнат в общежитии. Мы привели Стивена, и начали обсуждать это с ним, и тут возникла идея — „Все для разработчика“. О да. Все для разработчика, ведь нынче именно они делают королей. И он предложил такой тактический ход: „Эй! Почему бы вам не взять ноутбук Dell и не поставить на него Ubuntu?“».

«Я подумал: „Отличная идея, но в Dell этому не бывать“, — объясняет Бартон. — Пойду я в клиентскую группу — они мне скажут: „И сколько миллионов ты собираешься продать за первую неделю?“, а я отвечу: „Ну, о миллионах говорить не приходится“. В общем, я отбросил эту идею. А через 3–4 месяца в Dell был создан инкубационный фонд для поиска интересных идей среди сотрудников компании. Тут я был первым, кто ее высказал, и получил одобрение».

В движеньи, в движеньи...

«Изначально идея заключалась только в ноутбуке, однако судьба распорядилась иначе: за неделю до ее презентации в городе оказался Марк Шаттлворт [Mark Shuttleworth], — продолжает Бартон. — Он приехал и выступил; а поскольку мы продаем Ubuntu на ряде наших систем и я работал с Марком [раньше], у нас были хорошие отношения. Он нашел для меня время, я пришел к нему и спросил: „Ну, что ты об этом думаешь?“ Он ответил: „Мне нравится твоя идея. Но... будет реально круто, если мы сможем взять это все и соединить с облаком. Фактически, используя Juju, ты сможешь соединить это с облаком, и получится вся цепочка инструментов DevOps, и почему бы нам так не сделать?“ И я сказал, что мне это нравится».

«Мы вышли с этим вживую 7 мая (2012 г.) на Саммите разработчиков Ubuntu Developer в Окленде. Марк [Шаттлворт] поднялся и вышел с ним на сцену, — рассказывает Бартон. — Я не ожидал такого бурно-положительного отклика». В тот момент посещаемость блога Бартона просто зашкаливала. «Всегда было затаенное желание чего-то способного работать на симпатичном оборудовании, — говорит он. — Переломный момент наступил потом, когда мы сообщили, что запускаем по всему миру бета-программу, и подписались 6000 человек. Мы решили, что это дело хорошее — давайте пойдём и решим вопрос».

Как и с большинством оборудования ПК, пользователи Linux уже привыкли к вынужденным покупкам устройств с Windows, известных совместимостью с Linux, или переплыванию за ноутбук Apple, пригодный для Linux. Проблема с первым вариантом в том, что спецификации ноутбуков постоянно меняются; например, адаптеры Wi-Fi и Bluetooth меняются в последний момент, и вы не узнаете об этом, пока не принесете устройство домой и не попытаетесь установить на него Linux. Один из пользователей Linux, выбирающих опцию Apple — Линус Торвалдс, который берет с собой в поездки Apple Macbook Air с Fedora. Но в отличие от большинства из нас, Линус сам умеет создавать заплатки, чтобы обеспечить нормальную загрузку и работу оборудования, и он признает, что у Linux есть проблема с эффективным использованием появляющихся дисплеев Apple с высоким разрешением. В таком случае несколько удивляет в Sputnik то, что выбранное Dell оборудование не на 100% совместимо с текущим состоянием ядра Linux.

Спецификации оборудования

Само оборудование идентично Dell XPS 13, где редакция 2013 г. обновила 13-дюймовый экран до потрясающего разрешения IPS 1920×1080. Внутри — чипсет Intel Ivy Bridge с Core i7 i7-3537U CPU, настроенный на номинал 2 ГГц, но способный разогнаться до 3,2 ГГц, и 8 ГБ памяти DDR3, работающей на 1600 МГц. Intel HD Graphics 4000 обеспечивает видео. Наша модель включала твердотельный накопитель на 250 ГБ, загружающий рабочий стол Ubuntu всего за 8 секунд. И это — официальный продукт Dell, а значит, вы получаете официальную поддержку Dell, что, естественно, радует.

Дизайн — дело сугубо личное, но нам нравится стиль XPS 13. Корпус — 306 мм шириной, 205 мм длиной и 6–18 мм высотой. В общем, устройство производит впечатление премиум-класса, и корпус выглядит достаточно прочным, чтобы выдержать постоянные разъезды. Здесь нет боковых сгибов шасси, что мы всегда считали признаком того, что ваш ноутбук или построен, как танк, или имеет дизайн высшего качества.

» Пускай в нем и есть клавиша Windows, но на нижней стороне — логотип Ubuntu.

«Все для разработчика. Ведь нынче именно они делают королей.»

»



» Редакция 2013 имеет впечатляющий экран IPS 1920×1080.



► Корпус XPS 13 выглядит достаточно прочным, чтобы выдержать передрыги, связанные с постоянными разъездами.

Верхняя крышка и ободок алюминиевые, а дно сделано из неизбежного углеродистого волокна. Ультрабук достаточно легкий, чтобы нести его одной рукой в открытом виде. Экран покрыт стеклом повышенной прочности Gorilla glass, которое доходит до металлических краев. Рамка вокруг экрана всего 10 мм, и по нашему мнению, она выглядит намного изящнее, чем более широкая рамка MacBook Air.

Клавиатура полноразмерная; нам не очень понравился использованный для клавиш шрифт, но белая подсветка клавиш LED отлично продумана для работы ночью, хотя уровни яркости регулируются грубовато. И да, клавиша Windows по-прежнему на месте, но мы уже достаточно взрослые, чтобы не обращать на это внимания, тем более, что на нижней стороне красуется милый сердцу логотип Ubuntu. Еще важнее то, что, вволю попечатав, мы можем с полной уверенностью сказать, что по расстоянию клавиш друг от друга, утоплению клавиш и солидной основе клавиатуры отличная.

На машине только два порта USB: по одному слева и справа, и оба они USB 3.0, что налагает определенные ограничения. USB 3.0 будет работать только с относительно новыми ядрами, а наличие только двух портов означает частые поиски компромисса насчет того, что подключить, особенно если постоянно присоединен телефон для подзарядки.

Один из недостатков изящного дизайна — нет порта Ethernet. Наличие Wi-Fi — дело случая, и единственным выходом бывает соединение с проводной сетью — но с XPS 13 это невозможно, если только вы не носите с собой адаптер USB, что означает утрату порта USB. Единственный другой вариант — мини-DisplayPort.

Мы бы не стали упоминать о блоке питания, но небольшой кирпичик, который идет с XPS 13, не займет много места в вашем багаже, и на нем есть яркий LED на рабочем конце. Было бы удобно, если бы LED выключался, когда ультрабук заряжен, но эта функция передана оранжевому/белому LED на передней стороне корпуса. Аккуратная кнопка на противоположной стороне

► Стильный и минималистский дизайн означает минимум портов.



зарядника включает пять других белых LED — индикатор заряда батареи, даже при закрытом корпусе, и, наконец, можно использовать один из портов USB для зарядки вашего телефона или планшета, не пробуждая ультрабук ото сна.

Единственным обнаруженным нами недостатком оборудования было беспроводное устройство на беспроводном чипсете Intel Centrino Advanced-N 6235, с драйвером iwlwifi, которое работало нестабильно. Сначала оно соединялось с точкой доступа, но потом за несколько минут сигнал слабел до такой степени, что XPS 13 терял соединение. Даже с новейшим ядром и обновлениями Ubuntu 13.04 оборудование отказывалось работать. Удивительно, что это не проверили до февральского обновления, но еще более удивительно, что не работало беспроводное оборудование от Intel. Когда мы достали Dell жалобами, там заявили, что заменят все проблемные части оборудования устройством Killer Wireless-N 1202, которое должно решить все проблемы с драйверами. Даже в наихудшем случае заменить беспроводную карту mini-PCI не так уж трудно, однако должны ли вы этим заниматься, купив дорогой ультрабук, задуманный для Linux?

Поддержка ПО

Единственная официально поддерживаемая ОС — Ubuntu 12.04, ее-то вы увидите предустановленной. Выбор версии с долгосрочной поддержкой весьма разумен, и ничем не отличается от той ситуации, когда владелец Windows или OS X приобретает оборудование и не занимается обновлением операционной системы.

Вы будете получать обновления системы безопасности и заплатки до апреля 2017 года, и это достаточно долгий срок для любой ситуации, но не сможете установить новые релизы прило-

«Экран яркий, с насыщенными цветами и высокой контрастностью.»

жений, которыми пользуетесь, не используя сторонних архивов, загрузок бинарников и обновлений вручную. Как и следует ожидать, Dell 12.04 была дополнена специальным пакетным репозиторием для XPS 13, и ядро тоже специальное для XPS. В основном причина в том, что у тачпада до редакции Ubuntu XPS 13 не было драйвера Linux.

Бартон объяснил, как разрабатывался драйвер: «В написании сенсорного драйвера мы сотрудничали с Cypress. По сути, они писали его для нас и переместили выше этажом, так что он попал в ядро 3.9. И если оно у вас есть и вы хотите работать на Debian или CentOS — пожалуйста».

Драйвер и точно стал частью ядра: мы тестировали несколько дистрибутивов с XPS 13, и любое новое ядро работает без нужды в изменениях. Для Ubuntu 13.04 не требуется дополнительный репозиторий, поскольку, например, почти все заплатки были внесены в основной дистрибутив, и мы бы порекомендовали установить его вместе с более старым релизом LTS, чтобы получить доступ к новым пакетам.

В зависимости от настройки вашего тачпада, можно дополнить однократный щелчок и щелчок правой кнопкой касанием или касанием и нажатием на всю панель, а после установки ряда дополнительных пакетов мы также смогли добиться на этом устройстве достойной поддержки мультитач, хотя Linux по-прежнему сильно отстает от OS X в реализации управления жестами.

Еще одна область, в которой Linux, вероятно, не дотягивает до потенциала оборудования — экран. При плотности пикселей 165ppi (скачок со 118 более старых версий), втиснутых в диагональ 34 см, рабочий стол Unity нуждается в некоторой настройке для всех, чье зрение не стопроцентно, и здесь Linux выглядит несколько неадекватно. Шрифты часто слишком мелкие,

а границы окон трудно найти курсором мыши, особенно если надо менять размер окна. Все это можно переделать и приспособить, но настольному Linux, и в частности, рендерингу шрифтов, не хватает осознания плотности пикселей, работаете ли вы в Unity, Gnome или KDE. Правда, видеоборудование при родном разрешении отлично справляется с полноэкранным режимом, но поскольку оно относительно скромное, у нас возникли проблемы с играми 3D в полноэкранным режиме при средних настройках графики.

Звезда экранов

Когда вы перенастроите свой рабочий стол, экран выглядит невероятно: яркий, с насыщенными цветами и высокой контрастностью, а в сочетании с его разрешением, текст выглядит супер-четко, что отлично для редактирования. Вполне допустимо расположение окон бок о бок, например, для текстового редактора и web-браузера, благодаря высокому разрешению, и если вы любите открывать сразу несколько окон, вам понравится размер рабочего стола, который может отобразить экран. Для нас 1080p — отличная срединная точка для ноутбука такого размера. Экран XPS 13 блестящий, и нет опции сделать его матовым — хотя нам нравится обеспечиваемая им высокая контрастность, мы также знаем, что многие ненавидят блики. И у них это может вызвать отторжение. Другая мелкая проблема — затемнение экрана Ubuntu: оно выглядит просто фантастически, когда ваш сеанс работы засыпает, но вызывает дополнительную задержку при возобновлении работы.

Намного удачнее раздел экстренного восстановления Dell. Удерживайте комбинацию клавиш или выберите ее в меню загрузки Grub, и вы сможете восстановить настройку по умолчанию вашего ультрабука. Нам пришлось воспользоваться этим пару раз после того, как мы поэкспериментировали с другими дистрибутивами — даже с теми, которые добавляют собственные меню Grub — и каждый раз все отлично работало. Возможно, это и не самый важный момент, но в сочетании с отличной поддержкой, прилагаемой к ультрабуку, создается общее впечатление, что Dell точно знает, что делать, и понимает, что нужно на рынке Linux.

«Dell точно знает, что делать, и понимает, что нужно рынку Linux.»



Начальная цена за Developer Edition XPS 13 в Великобритании — £899 без НДС; это выше номинала, но дешевле, чем альтернатива Apple, вроде Macbook Air, которым пользуется Линус. Будь это обзор, мы были бы вынуждены признать, что есть много более дешевых ноутбуков, хотя все они явно больше и уродливее.

Но для профессионалов и путешественников цена окупается поддержкой и качеством самого оборудования.

Для нас важна надежда, что Developer Edition и дух

предпринимательства, стоящий за его разработкой, положили начало чему-то новому. И Canonical, и Dell проделали отличную работу с XPS 13, и, как обещал нам Бартон Джордж во время нашей беседы, это — будем надеяться — действительно только начало. **LXF**

► Раздел экстренного восстановления Dell особенно совершенен.

Dell Profile Tool

Спецприложение для воссоздания вашей среды разработки.

Dell сделала больше, чем просто dd Ubuntu на жесткий диск. Удобства ради, они нашпиговали дистрибутив массой пакетов, которые, по их мнению, могут понадобиться среднестатистическому разработчику. Но они пошли еще дальше, и создали то, что Dell называет Profile Tool. Идея заключается в том, чтобы избавиться от нудной установки пакетов для вашей сре-



► Dell надеется, что сумеет предложить профили разработки от высококлассных разработчиков — например, среда Python от Марка Шаттлворта или среда Android от Тима Брея [Tim Bray].

ды разработки, когда вы меняете компьютер, местоположение или даже компанию. Просто возьмите Profile Tool и разрешите ему заселить свою среду хранящимися в облаке настройками. Текущие профили включают Emacs, Ruby и JavaScript, и Бартону хотелось бы увидеть среди них профили с распределенным доступом от «звездных» разработчиков. Единственная проблема в том, что Profile Tool пока что в стадии разработки, и как таковой не слишком интуитивен или полезен. «Это будет один из тех моментов триумфа, ради которых мы рабски трудимся месяцами, чтобы потом выйти с победой, — сказал нам Бартон. — Мы собираемся сделать это публично».

«Эти части, они пока находятся не там, где нам хотелось бы их видеть, — признал Бартон. —

Про Profile Tool люди говорят: „Ну, я туда заглянул, и там кое-чего не хватает“. И это немного огорчает в проекте — нам жаль, что мы не продвинулись дальше! У нас есть ресурс за пределами Dell, который нам тут помогал, но не сработал так, как мы рассчитывали, и теперь, похоже, мы вернемся к обычной схеме разработки инженерами Dell. Нам просто нужно довести это все до нужной точки. Что бы была документация, чтобы люди могли просто взять и заработать. Мы хотим, чтобы сообщество приняло это на себя. Сейчас... у нас маловато документации; есть три профиля, и много народу уже проглотили кус, но мы все-таки хотим продвигаться немного дальше».

Познакомьтесь с проектом на его домашней странице на GitHub: <https://github.com/sputnik>.

Финал застолья

Алексей Федорчук завершает серию статей про рабочие столы, решив, что он добрался до конца истории.

Как было сказано в предыдущей статье (LXF176), к рубежу тысячелетий имело место доминирование KDE в ряду графических интерфейсов свободных UNIX-подобных систем. Однако KDE существовал вовсе не в вакууме, в чем мы скоро здесь и убедимся.

Обезжиренная среда

Буквально через несколько месяцев после выхода KDE появляется среда *Xfce*. Говорят, что первоначально название ее расшифровывалось как XForms Common Environment — разделяемая среда на XForms. К тому времени, как я ее впервые увидел, та же аббревиатура трактовалась как The Cholesterol Free Desktop Environment — «свободный от холестерина десктоп». Однако давно уже в *Xfce* не осталось и следов от библиотек XForms, да и того взрывного впечатления, как некогда, она уже не производит. Так что, подобно большинству аббревиатур FOSS-мира, ныне *Xfce* не расшифровывается никак.

Среду *Xfce* создал француз Оливье Фурдан [Olivier Fourdan], и не столько под впечатлением и по примеру KDE, сколько в попытке создания свободного аналога CDE. И в интерфейсе ее ранних версий четко прослеживается влияние первой рабочей среды UNIX'ов. Строилась она по совершенно другим принципам, чем KDE. Последняя стремилась стать самодостаточной средой со всеобъемлющим набором пользовательских приложений, а в *Xfce*, кроме средств управления окнами и сквозной настройки, имелся лишь очень небольшой комплект базовых утилит и приложений, типа файлового менеджера и эмулятора терминала. Поэтому *Xfce* был очень быстр и компактен.

Среда *Xfce* основывалась на наборе библиотек XForms весьма сложного происхождения,

но восходившем в конечном счете к *IRIS GL*, проприетарной графической библиотеке компании Silicon Graphics (ныне SGI). Да и сама XForms тогда не была свободной, а лишь бесплатной для личного некоммерческого использования (настоящую свободу в понимании FSF она обрела годы спустя). И потому отношение к ней дистрибьютелей было не менее осторожным, нежели к KDE. То есть ее существование игнорировали и Red Hat, и Debian. А поскольку по функциональности *Xfce* не сильно превосходила развитые оконные менеджеры, то и большого интереса энтузиастов она тоже не вызвала.

Отчетливо маргинальное положение занимала среда *Xfce* во время жизненного цикла своей 1-й и 2-й версии. Версию же 3-ю Фурдан весной 1999 года полностью переписал с использованием библиотек *GTK* — тех самых, которые, как мы скоро увидим, перед тем легли в основу среды Gnome. В результате этого «освобождения» *Xfce* стала включаться как дополнительный десктоп в ряд дистрибутивов, в частности, в ту же Mandrake и ее русскую редакцию. Однако дистрибутива, в котором она выступала бы десктопом по умолчанию, не существовало вплоть до середины нулевых, когда появились первые «легкие» системы быстрого развертывания, такие, как Zenwalk, где она пришлась очень ко двору. Но это уже переход к следующему этапу нашей истории.

А пока — несколько слов об *Xfce* версии 3. Именно в отношении ее название среды трактовалось «с уклоном в фитнес» — и к тому были все основания. Она работала чрезвычайно быстро и, так сказать, «импульсивно». Ощущение это передать словами почти невозможно, но оно осталось незабываемым — больше нигде и никогда мне такого испытать не довелось. Да и интерфейс ее тогда сильно отличался

от уже примелькавшихся к тому времени KDE и Gnome, не обнаруживая ни малейшего намека на «Windows-подобие», и скорее вызывая в памяти OS/2 эпохи «Кривого Мерлина».

Завершая этот раздел, еще раз подчеркну: появление *Xfce* не оказало большого влияния на мир рабочих сред для свободных UNIX-подобных систем. Ибо ось развития их проходила по линии противостояния KDE, о котором шла речь в прошлой статье, и Gnome, к рассмотрению истории которого мы сейчас перейдем.

Disclaimer, он же — отмазка

Однако прежде чем перейти к истории Gnome, я должен принести слова покаяния. Данная статья написана в рамках исторического цикла, в котором я, по мере возможности, обещал следовать завету Тацита — писать без гнева и пристрастия.

Увы, когда речь заходит о среде Gnome, мне этого завета придерживаться трудно. Разумеется, никакого гнева против этого десктопа я не испытываю, как и особого пристрастия. Одно время я был его пользователем. А все предыдущие годы честно (хотя и безуспешно) пытался найти те самые его несравненные достоинства, о которых столько говорили его приверженцы.

Нет, речь идет не о гневе или пристрастии, а скорее о чувстве недоумения: почему такая эклектическая, неорганизованная и даже не очень оригинальная в идейном плане среда завоевала столь широкую популярность. Разумеется, к тому были некоторые объективные предпосылки, о которых я скажу по ходу дела. Но во главе угла все-таки стоял субъективный фактор — тот самый, который можно назвать кредитом доверия пользователей. И который, как мы увидим, оказался фактически безграничным, хотя разработчики Gnome раз за разом подвергали его испытанию на прочность. Причем таким, на какие не решалась даже компания Microsoft, чья Windows даже на заре своего существования имела несоизмеримо большую пользовательскую базу. В общем, говоря об истории Gnome, мне очень трудно сохранять объективность — что и прошу учесть читателя.

Удар свободы

Однако вернемся к истории. Пока и KDE, и *Xfce* базировались на не вполне «идеологически выдержанных» библиотеках, истинные радетели свободы готовили удар по проприетаризму. Во главе этих радетелей встали Мигель де Икаса [Miguel de Icaza] и Федерико Мена-Кинтеро [Federico Mena-Quintero], основавшие в 1997 году проект Gnome (GNU Network Object Model Environment). Его можно рассматривать как ответ «твердых искровцев»... пардон, истинных Freesoftware'овцев прихлебным плюралистам из проекта KDE.

Целями проекта, насколько можно понять замысел его создателей, были (в порядке убывания приоритетов):

- 1 Создание полностью свободной, в терминах FSF, интегрированной рабочей среды.
- 2 Создание интегрированной среды, не похожей ни на одну из существующих сред.
- 3 Создание интегрированной среды, доступной для всех пользователей, вне зависимости от уровня их технической подготовки.



➤ Заря Xfce: пережитки проприетаризма в сознании.

Есть доля шутки

...в шутке про изменение мира. Одна бывшая наша соотечественница, уволенная из NASA чуть ли не по подозрению в шпионаже (в пользу русской мафии, надо думать), в поисках работы обращалась в том числе и в Gnome Foundation. Где на собеседовании ее спросили: хотите ли вы участвовать в изменении мира? Вопрос этот показался ей

смешным, и в итоге в этот самый фонд ее не взяли. Пришлось идти на работу в заштатную конторку, которая тогда называлась (да и сейчас называется) Google. А после капитализации последнего пришлось стать очень богатой женщиной. Так что политика Gnome подчас благотворно отражается на судьбах людей, с ней соприкоснувшихся.

Достижение первой цели требовало выбора библиотеки, на которой базировалась бы новая среда. А единственной по-настоящему свободной графической библиотекой в то время была базовая *Xlib* из штатной поставки XFree86, функциональности которой для построения интегрированного десктопа было явно недостаточно. Более развитые же библиотеки, о которых я упоминал — *Motif*, *XForms*, *Qt* — имели те или иные ограничения в использовании и распространении.

И тут на помощь пришел случай. Начиная с 1995 года, Спенсер Кимболл [Spencer Kimball] и Питер Маттис [Peter Mattis] вели разработку растрового графического редактора, который назвали *GIMP*, что первоначально означало General Image Manipulation Program (буква G приобрела значение GNU в 1997 году, когда эта программа удостоилась чести войти в данный проект). В ходе своей работы авторы программы столкнулись с острой потребностью в высокоуровневой графической библиотеке. Какую и создали, назвав ее *GTK* — *GIMP ToolKit*. Ее первая стабильная версия была выпущена весной 1998 года под архисвободной лицензией LGPL. И хотя библиотека эта создавалась с вполне конкретной, и достаточно узкой, целью, и в этом качестве показала себя самым лучшим образом, но как основа для интегрированной среды она не планировалась. Однако требование свободы оказалось сильнее технологических соображений, да и выбора особого у создателей Gnome не было. И это наложило отпечаток на все дальнейшее развитие среды.

Второй целью, не намного менее важной, нежели свобода, было создание среды, ни на что не похожей. По этой причине в качестве оконного менеджера Gnome был выбран *Enlightenment* — невероятно эффектный, но не очень удобный в настройке и использовании.

Так что третья цель — создание простой в использовании рабочей среды — по приоритетности оказалась где-то тридцать третьей. Да, первая версия Gnome, вышедшая весной 1999 года, фантастически красиво загрузилась с заставкой в виде анимированного театрального занавеса. Вот только настройка ее было не очень проста. Да до настройки обычно дело и не доходило. Ибо среда эта в те годы хорошо умела делать только одно: свопировать самое себя на винчестер. По крайней мере, при работе в Gnome индикатор его активности обычно не гас ни на мгновение.

Тем не менее, Gnome получил активную поддержку со стороны дистрибьютеров — тех самых, которые в штыхи встретили KDE и проигнорировали *Xfce*. Почти сразу после выхода первого релиза он стал десктопом по умолчанию в Debian'e (март 1999 года) и в Red Hat'e (апрель 1999 года). Не побрезговали Gnome и разработчики Mandrake, включив его как опциональную рабочую среду — именно в русской редакции этого дистрибутива я пытался ознакомиться с его первыми версиями.

Однако, насколько я могу судить, объявление Gnome умолчальным десктопом в Debian'e и Red Hat'e было скорее теоретическим (или, если угодно, политическим). Среди лично мне знакомых

в те годы дебианистов и редхатчиков пользователей этой среды я не припоминаю. Ну, а в Mandrake он вообще попал для галочки — чтоб был. И причина была в практической непригодности Gnome к реальному применению. Именно в те годы и говорилось: «KDE разработали обычные программы для удобства своего и пользователей. А Gnome создали крутые программисты для доказательства собственной крутизны». Надо сказать, что в те годы для такого утверждения были все основания. Да и ныне, во времена Gnome 3 и Gnome Shell, оно не выглядит совсем беспочвенным.

Великое противостояние

Разумеется, так продолжаться не могло: одно из знамен свободного софта практически не меняется его сторонниками. И разработчики Gnome вынуждены были, поступившись крутизой, перейти на путь упрощения. Началось все со смены оконного менеджера — место красивого, но непрактичного *Enlightenment'a* занял

«Развитие проходило по линии противостояния Gnome и KDE.»

непритязательный, но пригодный к употреблению *Sawfish*. А поскольку своего оконного менеджера у Gnome не было никогда, смена «управителей окон» с тех пор стала в этой среде традиционной.

Следующим шагом на пути упрощения стала 2-я версия Gnome, где простота настройки и использования были объявлены высшими приоритетами. А поскольку эталоном простоты в некоторых кругах тогда считалась Windows, в Сети прозвучал лозунг: «Сделать Gnome большей Windows, чем сама Windows». Правда, популярности он не снижался, и о нем постарались поскорее забыть. Но репутация среды, в которой может работать даже кухарка, за Gnome закрепилась прочно. Хотя функциональность его штатных приложений оставляла желать лучшего. А большинство хороших приложений, включавшиеся майнтайнерами в Gnome, разрабатывались независимо от него.

С другой стороны, KDE, при всех его многочисленных недостатках (а старым верным пользователям этого десктопа они были известны лучше, чем любым критикам), обладал тремя несравненными достоинствами:

1) обильным функционалом в отношении управления окнами и приложениями;

2) абсолютной настраиваемостью всего и вся;

3) набором штатных приложений и приложений от примкнувших сторонних разработчиков.

За средства настройки KDE подвергался критике. И местами справедливо: они были запутанными, интуитивно не очень понятными; настройки связанных, казалось бы, параметров часто оказывались раскиданными по разным пунктам меню. Но они были — и охватывали абсолютно все опции системы. Причем все это поддавалось настройке штатными средствами: необходимости в ручной правке конфигов практически не возникало.

В силу названных факторов в первой половине нулевых годов вокруг KDE сплотилось сообщество пользователей-прагматиков, нуждавшихся в качественном софте для решения своих задач в среде, подающей стопроцентную индивидуальную настройку. В том числе и пользователей, далеких от техники — юристов, переводчиков и прочих гуманитариев. Нельзя сказать, что таких не было среди пользователей Gnome — однако в его сообществе все-таки до некоторого момента доминировали компьютерные гики, вынашивающие планы изменения мира. Не случайно все опросы, касающиеся используемого десктопа, приносили в те годы чистую победу KDE — и с большим, подчас двухкратным, перевесом.

Правда, в таких опросах, как правило, не участвовали представители т.н. корпоратива, использующие, как правило, не тот софт, который хочется, а который прикажут. И потому картину эту нельзя считать совсем точной. Ибо с момента зарождения Gnome его интенсивно поддерживала самая мощная Linux-компания — Red Hat, в те годы бесспорный лидер корпоративного поля. А в 2003 году, после приобретения фирмы SUSE, выпускавшей одноименный дистрибутив, компанией Novell, за внедрение Gnome взялся игрок номер два. Ибо после расщепления этого дистрибутива на две ветки — коммерческую SLE (SuSE Linux Enterprise) и развиваемую сообществом openSUSE, в первой как десктоп по умолчанию приняли Gnome.

Почему — можно лишь гадать: ведь, как говорилось в **LXF176**, SUSE принадлежала к плотам KDE. В качестве одной из причин указывалось, что незадолго перед тем Novell прикупила компанию Ximian, занимавшуюся разработкой Mono — свободной реализации Microsoft .NET Framework. А поскольку основал Ximian Мигель де Икаса, технологии Mono были тесно завязаны на Gnome.

Не скидывая со счетов этого факта, можно предложить объяснение и попроще. Приобретя »

SUSE, Novell продвигала коммерческую ветку этого дистрибутива на корпоративный сектор американского рынка. Где, как уже было сказано, главным игроком была компания Red Hat со своим RHEL (Red Hat Enterprise Linux) и его десктопом Gnome. И Novell пришлось равняться на него.

Тем не менее, при несомненных успехах RHEL и SLE в корпоративном секторе, на пользовательских ПК результаты их деятельности по продвижению Gnome были малозаметны. А началом коренного перелома стал конец 2004 года, когда в мировом масштабе развернулось распространение дистрибутива Ubuntu как рабочей среды, по умолчанию использовавшей Gnome. Хотя предпосылки этого перелома закладывались еще ранее, году в 2002, в то время, когда Марк Шаттлворт только приступал к реализации своего проекта и принял судьбоносное решение — использовать в новом дистрибутиве не KDE, а Gnome.

Сам Марк объясняет это так: в то время в KDE были «одни рюшечки и менюшечки», а Gnome был простым и функциональным десктопом. Но тут он несколько лукавит: по функциональности KDE тогда на голову превосходил Gnome. А вот насчет простоты — что было, то было. Настройка KDE усложнялась от версии к версии (пропорционально росту функционала), а Gnome как раз в это время резко сменил вектор развития: раньше его позиционировали как крутейший десктоп от крутейших программистов, а теперь было решено сделать простейший десктоп, который может настроить и кухарка. А поскольку Ubuntu ориентировался (в том числе и) на эту категорию пользователей, выбор Gnome представляется вполне логичным. Но главным мотивом предпочесть Gnome, как мне кажется, было стремление выделить Ubuntu из ряда развивавшихся тогда (и, казалось, не без успеха) Систем Быстрого Развертывания (СБР), таких, как MEPIS, Xandros, Lindows: все они, базируясь, как и Ubuntu, на Debian'e, выбрали десктоп KDE.

Так или иначе, но вследствие сочетания обоих факторов — массовой бесплатной рассылки дисков с Ubuntu и «упрощения» Gnome — Gnome пошел в народ. Причем первый фактор был, безусловно, весомее: в ряды линуксоидов влилось пополнение, которое ничего, кроме Ubuntu и Gnome, в глаза не видело, и для него эти понятия стали, по крайней мере по первости, тождественны понятию Linux. Однако и второй фактор не сбросить со счетов: покажись Gnome неопитам слишком сложным, они бы выбросили диски с Ubuntu на помойку без всякого сожаления, ибо халява. И так, расчет Марка был точен. Хотя позднейшие события показали неоднозначность его последствий.

От KDE 4 до Gnome 3

Следующее судьбоносное событие, повлиявшее на распределение пользователей по десктопам (или десктопов по пользователям?) произошло в начале 2008 года. Это был выход KDE 4.0. Данная версия KDE разрабатывалась примерно столько же времени, сколько хватило, чтобы смениться KDE 1, KDE 2 и KDE 3. И в анонсах на ранних стадиях разработки обещались если и не золотые горы, то по крайней мере серебряные. В частности,

Слово — не воробей...

Сейчас, по прошествии нескольких лет, становится понятно, что разработчики KDE почти все сделали правильно. Допустив лишь одну маленькую тактическую ошибку: они обозвали релизом достаточно сырую тестовую версию. И никакие объяснения, что это не совсем-таки настоящий релиз, не в силах

были преодолеть магию Слова. Хотя, с другой стороны, не пойдя KDE'шники на такой отчаянный шаг, разбудивший здоровую рабочую злость и пользователей-тестировщиков, и сторонних разработчиков — кто знает, может быть, 4-я версия KDE отлаживалась бы и по сей день в тестовом режиме.

там содержались толстые намеки на то, что KDE 4 будет обходиться без Иксов (хотя каким образом — не уточнялось). Правда, со временем анонсы становились все скромнее (в частности, из них исчезло упоминание об отказе от X-сервера), но по-прежнему оставались многообещающими.

Тем большим было разочарование, я бы даже сказал, шок, по выходе релиза KDE 4.0. Вместо стабильной, функциональной, настраиваемой среды с привычным интерфейсом мы увидели нечто глюкавое и падучее, без половины функций и настроек. Причем уменьшение вариантов конфигурирования вовсе не сопровождалось его упрощением — напротив, оно стало еще более запутанным. Но зато все это было обвешано, как новогодняя елка, всякими красотами в виде плазмойдов. Надо сказать, что разработчики KDE 4.0 честно предупреждали, что эта версия предназначена еще не для практического применения, а лишь для ознакомления с новыми возможностями и их тестирования. Но сделали они это достаточно невнятно, и майнтейнеры дистрибутивов дружно бросились включать ее в свои новые релизы (обычно, правда, параллельно с предыдущей, 3.5.X). А пользователи, приняв все это за чистую монету, столь же дружно стали «нулевыми четверку» устанавливать.

Результат оказался более чем предсказуем. Поскольку было вполне очевидно, что дни 3-й ветки сочтены, начался массовый отток пользователей KDE на запасные аэродромы. Для кого-то им стал Xfce, кто-то вернулся ко временам информационной юности — на менеджеры окон. Но немало бывших KDE'шников попросило политического убежища в стане Gnome.

Тут выяснилось, что в Gnome образца 2008–2009 годов жить можно. Конечно, огорчал подход к конфигурированию среды, когда в интерактивном режиме был доступен лишь самый минимум, необходимый, по мнению разработчиков, «просто» пользователю, а для реализации более сложных (по их же представлениям) опций следовало лезть в нечто вроде реестра, еще менее интуитивно понятного, чем опции настройки KDE, и гораздо более бедного. Видимо, разработчики Gnome руководствовались старым советским принципом: «Что позволено папарту — не позволено бичу.»

Поначалу удручало убожество штатных приложений — например, *Gedit*'а против *Kate* или *Nautilus*'а — против *Konqueror*'а старого образца. Однако тут нашлись и плюсы: в Gnome органично интегрировались любые сторонние приложения, основанные на *GTK*. И *Gedit* для всамделишной работы легко заменялся *Geany*, а *Nautilus* — дополнялся *PCManFM*'ом. Кроме того, Gnome органично интегрировался с такими дистрибутивами, как Fedora. Так что на рубеже десятилетий между KDE и Gnome был достигнут примерный паритет. Хотя KDE и сохранял некоторое преимущество — за счет а) стойких его приверженцев и б) энтузиастов новаторства. Тем более, на месте он не стоял, а постепенно допиливался до юзабельного состояния, сам по себе, во-первых, и портированием на 4-ю ветку «трешечных» приложений — во-вторых. Я, утратив в то время всякий интерес к KDE, не следил за этим процессом. И к версии 4.4 KDE стал вполне похожим на настоящий как сам по себе, так и в отношении своих штатных компонентов.

Тем не менее, отток пользователей от KDE казался необратимым, и Gnome утвердился с ним на одной ступеньке пьедестала почета, ну, может быть, на полступеньки ниже. Повторяю, заслуги самого Gnome в приближении к вершине пьедестала были минимальны, ибо главные роли в этом спектакле сыграли упомянутые факторы — массовое распространение Ubuntu и ошибки команды KDE в продвижении новой версии своей среды. Добавим сюда неожиданный рост популярности дистрибутива Fedora, тесно интегрированного с Gnome. Причем рост этот начался не благодаря Gnome, а скорее вопреки ему: многие пользователи готовы были мириться с этой средой ради достоинств самого дистрибутива. Который как



▶ Ранний Gnome: первый шаг к простоте.

раз тогда время временно отказался от своей роли «тестовой площадки» коммерческого RHEL'a и обратился лицом к пользователю.

Сказанное выше касалось распространения Gnome в мировом масштабе. А в нашей стране дело его укрепил успех проекта Russian Fedora, воплощенный в сборках RFRemix, кардинально улучшавшихся от версии к версии, вплоть до RFRemix14 — апофеоза отечественного дистрибутива. Который (и это не только мое мнение) можно было смело рекомендовать русскоязычным пользователям любой сферы деятельности и почти любого уровня подготовки. Казалось бы, вот он — тот самый готовый к десктопу Linux, разделенный человеческим лицом в виде интегрированной среды Gnome. То есть то самое, о чем говорили со времен первой версии Mandrake с его KDE. Однако — увы, это только казалось.

Ибо свежий ветер перемен затронул и Gnome, обрушившись с ураганной силой: в апреле 2011 года принес он с собой Gnome 3 с его Gnome Shell. Если и раньше, во времена Gnome 2, среда эта не утомляла своего пользователя избытком настроек, даже в собственном реестре, то с переходом к 3-й ветке настройки по сути исчезли как класс — или позднее, в очень ограниченном количестве, стали доступны благодаря сторонним утилитам. Интерфейс же выглядел настолько революционным, что от привычных рабочих сред, казалось бы, не осталось просто ничего. Однако большинство нового — это хорошо испорченное старое. Так, оверлейный режим Gnome Shell'a вызывал в памяти времена, когда не было еще фиксированных множественных рабочих столов, но был зато единый рабочий стол с виртуальным разрешением в два и более раз выше физического (см. LXF173). А вертикальный ряд пиктограмм вдоль левого края экрана выглядел как реплика Window Maker'a, который, в свою очередь унаследовал эту идею от NeXTSTEP'a (см. LXF174).

Неизвестно, позаимствовали разработчики Gnome Shell'a эти идеи у предшественников или изобрели свои велосипеды сами. Но в любом случае, ничего ультра-революционного в плане интерфейса эта среда не содержала. Разве что слияние виртуального рабочего стола с вертикальными пиктограммами. Получилось бы, возможно, вполне удачное нишевое решение — скажем, для планшетов и прочих устройств с сенсорным экраном. Да вот беда, никто из производителей подобных гаджетов внедряя Gnome 3 не спешил: нишу эту прочно оккупировал Android. А для обычных ПК или ноутбуков Gnome 3 был малопригоден. По крайней мере, для тех применителей, кто на них работал, а не развлекался.

Однако разработчики Gnome Shell'a с этим не согласились. И попытались представить свое нишевое решение как универсальное. Причем с такой агрессивностью, с какой я не сталкивался за полтора десятка лет соприкосновения с миром Open Source. На любую критику в адрес этой среды у ее фанатиков (не фэнов, не фанатов, а самых натуральных фанатиков) существовал универсальный ответ: все это гениально и прогрессивно, а кто того не понимает — ретроград и обскурант.

Было среди приверженцев Gnome и другое течение — стратотерпцев. Не будучи модернистами, но сохраняя склонность к Gnome, они предпочли смириться и привыкать к новомодному Gnome Shell'у, по принципу «стерпится — слюбится». Надо сказать, не все они были «терпилами»: некоторые активно создавали всякого рода твичеры для настройки «третьегонома» — то есть то, чем его создатели пренебрегли, вполне цинично возложив эту миссию на «самых утопающих».

Наконец, возникло и третье течение — приверженцев старого и доброго Gnome, не желавших привыкать к Gnome новому и недоброму. И они пошли своим путем, да не одним, а сразу двумя.

Первый путь был ультраконсервативным: создание форка Gnome 2, названного Mate. В отличие от близкого по идее проекта Trinity KDE (см. врезку), этот проект был начат практически сразу по выходе «третьегонома», когда запас спортивной злости пользователей его 2-й версии еще не был растерян в дискуссиях на форумах и блогах. И потому Mate развивался (и развивается) вполне успешно. Второй же путь был более радикальным. Ряд пользователей 2-го Gnome, понимая, что для него слова, некогда увиденные царем Вальтасаром, уже написаны, занялись стилизацией интерфейса Gnome 3 под внешний вид прошлой версии. И в общем тоже не без успеха. Проект этот, получивший имя Cinnamon, быстро перерос исходные рамки. Это уже не история, а современность, и о ней вкратце рассказано в колонке к Обзорам.

Здесь уместно было бы сказать и о среде Unity — однако ее история, как и современное состояние, были достаточно подробно рассмотрены в статье «Феномен Ubuntu» (см. LXF173).

А что же другие?

Этот фрагмент истории десктопов был начат кратким рассказом о Xfce, и ею же уместно его и завершить. Мы оставили ее на уровне 3-й версии, которая, будучи переписана на основе библиотеки GTK, обрела статус свободной среды. Но на этом ее история не закончилась. При подготовке релиза 4.0 Xfce снова была полностью переписана — теперь на основе следующей, 2-й, версии GTK. И не просто переписана изнутри, а кардинальным образом сменила интерфейс. От былого сходства с KDE не осталось и следа — внешности ее по умолчанию было придано Gnome-подобие, а со временем, в версии 4.6, она обрела и своего

рода реестр xfconf, также представляющий собой аналог GConf из среды Gnome.

Потеряла Xfce и свой «темперамент», хотя оставалась достаточно «легкой» и быстрой (впрочем, по первым двум пунктам можно найти и возражения). Ее компактность трудно подвергнуть сомнению. И благодаря этому она стала десктопом по умолчанию ряда систем быстрого развертывания, строившихся по принципу «одна задача — одно приложение». Среди таких систем наиболее известны, пожалуй, Zenwalk и его родич — Salix. А затем она была задействована в Xubuntu, одном из представителей семейства Ubuntu. Хотя сборки с Xfce нынче можно видеть и в большинстве «больших» дистрибутивов, таких, как Fedora, openSUSE, Debian и ряде других.

Но, как я только что сказал, разрастание Xfce вызывало нареkania — хотя, на мой взгляд, не вполне обоснованные. Тем не менее, в противовес ей были созданы еще более «легкие» десктопы — сначала LXDE, а затем Razor-qt, основанные на библиотеках GTK 3 и Qt, соответственно. Ныне их разработчики обозначили намерение слить оба проекта воедино. Но здесь мы уже окончательно покидаем историю и переходим к современности.

Вместо заключения

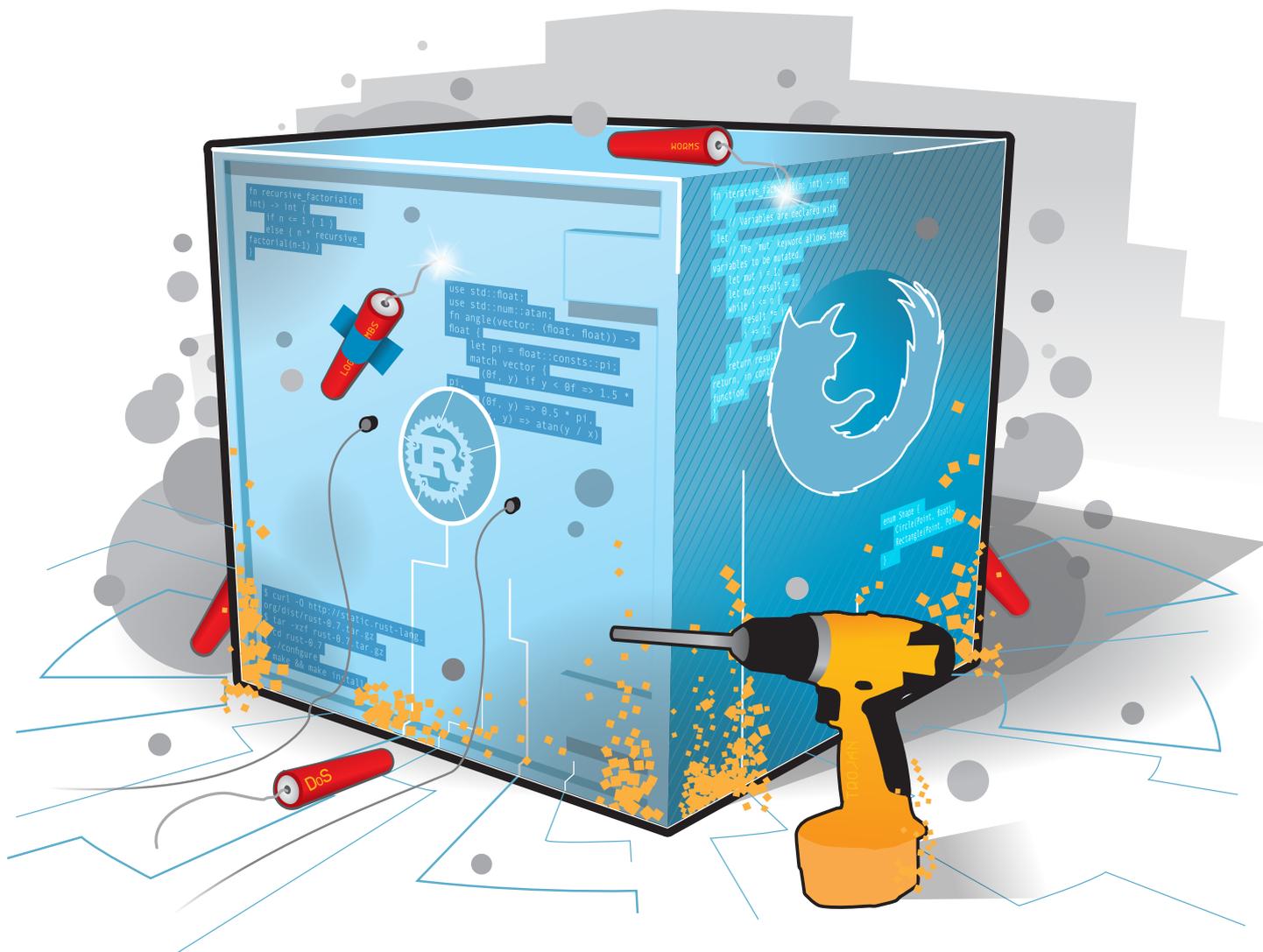
Этой статьей я завершаю затянувшийся цикл, посвященный истории UNIX, Linux и их графических интерфейсов. В нем осталось немало пробелов — например, я почти не затронул историю ряда BSD-систем, типа NetBSD и OpenBSD, сыгравших немалую роль в становлении мира свободного софта в том виде, в каком он существует сейчас. Не говорилось и об истории приложений для UNIX и Linux, хотя некоторые ее аспекты были весьма драматичны — например, эволюция StarOffice, превратившегося в современные Libre Office и Apache Open Office. Немало поучительного было и в истории других офисных пакетов для UNIX и Linux, начиная с первых из них, Applix и AbiSuite (реликт последнего — современный AbiWord). Весьма поучительна была попытка портирования на Linux пакета Corel Office, включавшего, среди прочего, и легендарный WordPerfect. Возможно, кто-нибудь восполнит лакуны в моем повествовании. Мне же остается только вспомнить слова Козьмы Пруткина про объятие необъятного и попрощаться с читателями. Возможно — до тех времен, когда историей станут события современности. LXF

Попытка не удалась

Была и попытка реанимации KDE 3 в виде, привычном для пользователей этой ветки (и всех ей предшествовавших версий). Это проект Тимоти Пирсона [Timothy Pearson] под названием Trinity KDE. Он представлял собой дальнейшее развитие 3-й ветки с рядом заимствований из 4-й — тех, что не шли в разрез с исконной идеологией KDE.

Увы — начатый во второй половине 2010 года, он явно запоздал. За два с половиной года, прошедшие с момента выхода KDE 4.0, эта среда

в своем традиционном виде растеряла изрядную часть своих некогда преданных поклонников: одни, скрепя сердце, погрузились в мир плазмойдов 4-й ветки, другие же мигрировали на более иные десктопы. А новые пользователи воспринимали «четверку» как данность — тем более, что к этому времени она стала пригодной к употреблению. И Trinity оказалась просто невостребованной широкими народными массами. Хотя некоторое число поклонников и обрела.



Что за штука...

Rust?

Марко Фиоретти представляет язык программирования от Mozilla, способный расширить возможности Интернета и защитить от взломщиков.

В Ну и сразу вопрос на миллион: что за штука — Rust?

О Rust (www.rust-lang.org) — это открытый язык программирования, разработкой которого занимается Mozilla Foundation.

В Язык программирования? Тогда для начала объясните, пожалуйста, почему 'Rust' [англ. ржавчина]? По-моему, название дикое.

О Официальных комментариев нет, по Сети ходят лишь версии. Самая популярная в том, что Rust предназначен не для хромирования — блестящих внешних компонентов приложений Mozilla — а для внутренней структуры сетевого ПО.

Другое вероятное объяснение — что в Rust намеренно используются принципы и теории

программирования, по меньшей мере, 10-летней давности. Другими словами, Rust стремится решить актуальные проблемы, опираясь на наиболее подходящие для этого идеи и методы, пусть даже появившиеся лет 30 назад, но не снижавшие популярности из-за недостатка внимания к ним.

В Что ж, похоже, что по такой логике, Rust — неплохой вариант. Но, уходя в сторону от имени, зачем вообще нужен новый язык?

О Потому что при текущих нагрузках современным Интернет-серверам и клиентам требуются качества, не являющиеся высокоприоритетными в языках системного программирования, таких как C или C++. Если говорить конкретнее, Rust предназначен для крупных клиент-серверных

приложений, работающих в сети Интернет, в условиях длительной нагрузки, и свободен от рисков безопасности и сбоев производительности, которые можно встретить в других языках.

В Звучит, по меньшей мере, амбициозно. Как воплотились эти требования и заявки на практике?

О Rust пытается быть ближе к тем, кто программирует на C и C++. Отчасти потому, что он призван, за большинством случаев, вытеснить последние. Так что обеим сторонам имеет смысл взаимно уподобить функции, облегчив разработчикам этот переход. Параллельно, Rust с удовольствием возвращается к использованию методов из многих других языков, таких как Erlang и Go

(еще один язык Google, подробнее смотрите на <http://golang.org>), совмещающая их и создавая новые.

В Erlang — это же тот язык, на котором создается ПО для крупных АТС, так?

О Да, все верно. Параллелизм и подобные функции (в различных формах) есть и в Rust, и в Erlang. Однако Erlang — все же другой биологический вид, его в АТС, в обозримом будущем, Rust (как и что-либо еще) не сменит. Потому что, помимо всего прочего, «высокая доступность» в сфере связи и в сфере Интернет понимается по-разному.

В А как насчет Go? Это более прямой конкурент Rust?

О Не совсем. Хотя Go и предназначался для написания параллельных интернет-приложений, он решает другие задачи, и иначе, чем Rust. По мнению некоторых программистов, хотя Rust может заменить C и C++ в крупных проектах, Go является наиболее простым и подходящим вариантом для серверного ПО, а также других программ и утилит, которые сегодня пишутся на Python или Java.

В Каковы, по вашему, основные функциональные возможности Rust?

О По большому счету, не углубляясь в детали, понятные только разработчикам, я бы выделил высокую безопасность и производительность, а также параллелизм.

В Хорошо, давайте тогда с параллелизма и начнем.

О Иногда параллельное ПО определяют как способное «плодиться» и одновременно запускать множество независимых копий собственных частей, способных взаимодействовать друг с другом. Отметим, что здесь действует совершенно иной принцип, чем когда, допустим, веб-сервер одновременно взаимодействует со множеством независимых пользователей! В Rust могут параллельно выполняться тысячи или миллионы легких задач [lightweight tasks], без риска, что крах или зависание одной из них обрушит все приложение, или что единичное вредоносное проникновение поставит под угрозу взлома все личные данные.

В Каким образом будут обеспечиваться высокая безопасность и производительность?

О Rust затрудняет создание вредоносного кода одним простым способом: чтобы получить тот же уровень «свободы действий», что и в C, вам нужно запросить разрешение, и далее еще поработать, чтобы это стало возможным. Таким образом, Rust должен стать гораздо более безопасным и надежным эквивалентом C, при тех же функциональных возможностях.

В То есть в Rust функций намного меньше, чем в других языках?

О Нет. В Rust есть и структуры, и указатели, и все прочее, к чему привыкли современные программисты. Просто в нем, по умолчанию, гораздо больше ограничений. Примером может послужить так называемый «контроль изменчивости [mutability control]».

В Да уж, звучит как коварная программистской инженерии; но я полагаю, что это нечто другое...

О Да, действительно, похоже на какой-нибудь замысел Umbrella Corporation [вымышленная корпорация, играющая ключевую роль в сюжете серии видеоигр *Resident Evil*, — прим. пер.], но расслабьтесь. Имеется в виду, что в отличие от многих других языков, переменные Rust — вовсе не переменные. Чтобы изменить одну из них, в процессе работы программы, нужно объявить это, используя ключевое слово `mut`:

```
let mut my_initial_salary = 1000
```

Другие характеристики, делающие Rust более безопасным, чем, скажем, C или C++ — это параметризация типов и отсутствие null-указателей.

«Rust предназначен для крупных клиент-серверных приложений в сети Интернет.»

В Погодите! Вы сказали, что не будете бросаться терминами, понятными только программистам!

О Ну, соврал; но не переживайте. Невозможно представить язык программирования, не вдаваясь в некоторые детали, но все гораздо проще, чем кажется.

Параметризация типов, например, означает, что Rust позволяет, или скорее заставляет вас очень точно определять (то есть параметризовать) ваши основные типы и функции, избегая необходимости постоянно приспосабливать (то есть назначать) несколько стандартных типов переменных.

В А как насчет упомянутого отсутствия null-указателей?

О Rust запрещает использование указателей (переменных, содержащих адреса других переменных) со значением NULL (если только вы не упростили Rust это сделать). Как бы банально это ни звучало, одно такое ограничение в коде избавляет от целого класса ошибок, прежде омрачавших жизнь программистов на C и C++.

В Похоже, что управление памятью в Rust — невероятно сложное занятие...

О Да уж, друг мой, не простое, так что я вам объясню буквально в двух словах. Для начала, заботу о том, когда распределенная память не востребована, Rust не доверит вам, а оставит себе. В нем есть механизмы, разрешающие доступ к ячейке только в рамках только одной задачи и автоматически освобождающие всю занимаемую память после ее завершения. В другом случае, вы можете «зарезервировать» ячейки для нескольких задач, предотвратив при этом параллельный доступ, например, в случае состояния гонки или повреждения данных.

В Правда? У меня сложилось впечатление, что разделение данных между разными задачами в Rust вообще невозможно.

О На самом деле, так и есть, если только вы не пометите все наиболее разгульные части кода как «небезопасные». Такие могут использоваться в памяти в более произвольных (то есть опасных) формах.

В Но если программисты по-прежнему смогут писать ненадежный код, какой смысл использовать Rust?

О Смысл как минимум в том, что как исходный разработчик, так и все остальные, кто будет впоследствии поддерживать этот код, немедленно узнают, какие части огромного приложения могут оказаться проблематичными.

В А как, на практике, Rust может помочь начинающему программисту сделать свой код безопасным?

О Самым суровым, но эффективным способом: за счет архитектуры и встроенных ограничений, о которых я только что рассказал, компилятор Rust может выявить гораздо больше ошибок и уязвимостей, чем C или C++, и будет клевать вас до тех пор, пока вы их не исправите, иначе вам будет нельзя даже запустить свое приложение.

В Остался один вопрос: когда мы сможем увидеть приложения Rust в Интернете?

О Текущая версия Rust, 0.7, выпущена 3 июля 2013 года. Но независимо от номера версии, Rust — еще очень молодой язык и гораздо менее сложившийся и протестированный на практике, чем тот же Erlang. Так что могу сказать одно: будем надеяться, это будет где-нибудь в 2014. Чтобы быть в курсе последних новостей, обратитесь на официальный сайт: www.rust-lang.ru. **LXF**



По рецептам доктора Брауна

Д-р Крис Браун

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

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

Мир в проводах

Недavno моей старой подруге Мэри прислали квитанцию штрафа за неправильную парковку. Уверенная, что она ничего не нарушила, Мэри поговорила по телефону с приятной дамой из нашего муниципального совета, и та предложила ей отправить апелляцию через сайт муниципального совета. Тут в этой истории возник я: Мэри понадобилась помощь в заполнении формы. Какая? У Мэри не было компьютера.

Эта история добавила боеприпасов в обойму моей жены, которая постоянно борется с «опутыванием мира проводами» и считает, что компьютерная грамотность нашего населения и его возможность подключиться к Сети оцениваются весьма вольно. Она чувствует, что в мире начинается дискриминация тех, кто не может или не хочет пользоваться Интернетом. Если верить моей жене, у большинства людей нет доступа в Интернет. Думаю, это объясняется в основном демографией ее знакомых (т. е. людей пожилых), ибо по данным Службы национальной статистики в 2012 году доступ в Интернет из дома был у 80% жителей Великобритании. Но остается 20% (5 млн семей) без него, и точка зрения жены пока актуальна.

Взять, например, мою тещу. Ей 90, и она считает Google персонажем «Властелина колец». Как и многие другие, она все еще ездит в автобусе, сжимая в руке пару купюр, чтобы заплатить за телефон в почтовом отделении.

Должны ли мы продолжать помогать тем из нас, кто не живет поверх стека TCP/IP? Или стоит подождать 10–20 лет, пока текущее поколение луддитов не состарится и не умрет? chris.linuxformat@gmail.com



Дух сообщества

Компании-разработчики ПО рождаются, покупаются, продаются и умирают. С открытым ПО все иначе...

Энтузиасты Linux любят слушать, как проприетарные компании становятся открытыми, и таких компаний много. Но программы, которые изначально были открытыми, тоже нередко становятся проприетарными, и то, что происходит после этого, бывает интересно...

Иногда изначальные разработчики создают свою компанию. Классический пример — PGP (Pretty Good Privacy), программа, широко используемая для шифрования почты. Изначально ее написал Фил Циммерман [Phil Zimmerman], потом

он основал PGP Inc, которую позже приобрела Network Associates, которая продала ее PGP Corporation, которая была куплена Symantec. Просекли? В это же время открытое сообщество сплотилось и создало GPG, альтернативу с лицензией GPL, соответствующую спецификациям OpenPGP.

Еще один хороший пример — программа обнаружения вторжений *Tripwire*. Изначально она была написана в Университете Пэрдью [Purdue University], затем один из ее разработчиков основал *Tripwire Inc* и получил права. Они сделали код, написанный в университете, доступным (*Open Source Tripwire*), но затем все хорошие нововведения (*Tripwire Enterprise*, *Tripwire Log Center*) стали закрытыми. Сообщество, опять же, ответило на это программой *AIDE* с лицензией GPL.

К менее удачным примерам относится OpenSolaris от зачинателя Unix Sun Microsystems, сообщество разработчиков которой быстро распалось, когда Oracle купила Sun в 2010-м. Ответом сообщества стала Illumos Foundation, которая сделала разветвление кода и создала OpenIndiana.

Хотя я не собирался делать Oracle козлом отпущения в этой статье, она принимает участие и в двух последних примерах. В начале нулевых *OpenOffice* стал представлять серьезную конкуренцию Microsoft. Но Oracle, купив Sun, сразу свернула его. В данном случае у него было минимум два последователя: *Libre Office* и *Apache Open Office*. Наконец, по тому же пути пошел и *MySQL*. Ответом сообщества стала *MariaDB*.

В этих историях ярким светом сияет стойкость «сообщества». Империи Oracle, Microsoft и Google рухнут, а открытое ПО все еще будет жить.



► Illumos — феникс, восставший из пепла Solaris — иллюстрирует возможности открытого ПО к восстановлению.

Пишите письма

Если у вас есть история о том, как сообщество «подшло к калитке», чтобы спасти программу, которая исчезла за дверями закрытого ПО (в особенности если вы сами участник этой истории), я был бы счастлив ее выслушать. Может

написать мне по адресу: Chris Brown, c/o Linux Format, Future Publishing, 30 Monmouth Street, Bath, BA1 2BW, UK/Великобритания, или же сбросить электронное письмо на chris.linuxformat@gmail.com.

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ВИРТУАЛИЗАЦИЯ: ЭКОНОМИЯ НА СТОИМОСТИ IT-ИНФРАСТРУКТУРЫ ДО 90%



Единая инфраструктура
на базе свободного
программного обеспечения

Нет лицензионных платежей —
расходы только на внедрение
и техническую поддержку

Минимальные затраты
на оборудование
за счет виртуализации



[www.linuxcenter.ru/shop/
linux-software/office/kitezh](http://www.linuxcenter.ru/shop/linux-software/office/kitezh)

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

Радости Awk

Awk Сколько можно понаделать одной строкой кода? Доктор показывает нам свой любимый малый язык...

Для начала разберемся со странным названием. Awk назван по фамилиям троих его изобретателей — Альфреда Ахо [Alfred Aho], Питера Вайнбергера [Peter Weinberger] и Брайана Кернигана [Brian Kernighan]. Для меня Awk — это квинтэссенция Linux. Почему? Ну, это текстовый фильтр, и им исключительно удобно пользоваться в каналах, но главное — в нем можно делать Реально Полезные Вещи всего несколькими строками кода.

Awk — полноценный язык программирования. В нем есть переменные, арифметические операции, массивы, циклы, ветвление, функции и все прочее, что языку программирования и полагается. Но здесь я не буду пытаться составить систематическое описание языка; моя цель — показать и объяснить несколько примеров того, как можно делать полезные вещи одной-двумя строками кода. Торжество тезиса «Прекрасное в малом», если хотите.

Основной инстинкт

Основной инстинкт Awk — считать входной поток строка за строкой и разбирать строку на поля. В программе Awk для обращения к полям используются обозначения \$1, \$2 и т.д. По умолчанию поля разделяются пробелами, но это можно изменить, в чем мы вскоре убедимся. В примере на рисунке показан структурированный «список покупок» и его разбор на поля.

Программа на Awk состоит из набора шаблонов [patterns] и действий [actions]:

```
pattern { action }
pattern { action }
```

➤ **Основное действие Awk — разбиение строк ввода на поля. NF — количество полей, NR — количество строк.**



Шаблон выбирает строки для обработки, а действие описывает, что будет сделано со строками, соответствующими шаблону. Так как пока мы не знаем, на что похожи шаблон и действие, смысла в этой информации немного, но вот простой пример. Эта команда Awk покажет точки монтирования (второе поле) для всех строк в `/etc/fstab`:

```
$ awk '{ print $2 }' /etc/fstab
```

Здесь `{ print $2 }` — действие (печать второго поля). Где же шаблон, спросите вы? Хороший вопрос! Его нет, и в данном случае (весьма распространенном) Awk будет выполнять действие над каждой строкой.

Если вы попытаетесь выполнить команду со своим `fstab`, то скорее всего получите некорректный результат из-за строк комментариев. Строки комментария начинаются с `#`, и с помощью шаблона (в данном случае, регулярного выражения, обрамленного слэшами) можно отфильтровать эти строки таким образом:

```
$ awk '/^[^#]/ { print $2 }' /etc/fstab
```

Если вы не владеете языком регулярных выражений, поясню, что регулярное выражение (между слэшами) говорит «строки, которые начинаются с любого символа, кроме `#`». Вот еще один пример выбора одного поля. Вывод команды `date` имеет следующую структуру:

```
Mon Nov 4 19:12:42 BST 2013
```

и мы можем вывести поле со временем так:

```
$ date | awk '{ print $4 }'
```

Здесь важно отметить, что Awk обрабатывает поток данных из канала, а не файл. (В данном случае в потоке всего одна строка — в примерах Awk это бывает довольно часто). Отметим очевидный (но важный) момент: чтобы обрабатывать текст с Awk, нужно понимать структуру данных, в частности, какие данные содержатся в каждом поле.

В Linux есть много структурированных текстовых файлов, где в качестве разделителя используется : — на ум приходит `/etc/passwd` — поэтому иногда нам нужно указать Awk на использование другого разделителя, как в этом примере, в котором выводятся только имена пользователей (первое поле) из файла паролей:

```
$ awk -F: '{ print $1 }' /etc/passwd
```

Обратите внимание на использование параметра `-F` для указания разделителя полей. (Существуют и другие способы разделения полей, в том числе поля с фиксированной длиной и более хитрые методы на основе регулярных выражений.)

Сделаем еще шаг вперед и выведем только имена пользователей, соответствующие учетным записям обычных пользователей. В Red Hat их можно определить по универсальному идентификатору (третье поле), он должен быть больше или равен 500, и выбрать их в шаблоне Awk можно так:

```
$ awk -F: '$3 >= 500 {print $1}' /etc/passwd
```

Этот с виду невинный пример на самом деле дает первый намек на истинную мощь Awk: он способен выполнять арифметические операции, что не по зубам средствам вроде `grep` и `sed`.

Вот еще одна однострочная программа, выполняющая арифметические действия. Она показывает среднее для чисел, введенных в строке:

```
$ awk '{ sum=0; for (i=1; i<=NF; i++) sum += $i; print sum/NF; }'
```

```
4 6 14
```

```
8
```

```
72 123 45 17
```

Подстановка команды

Подстановка команды — прием программирования в оболочке, который напрямую не относится к Awk, но часто используется с командами Awk. Предположим, у меня есть файл `hostlist`, содержащий список имен компьютеров, по одному на строку. Строка

```
$ ping `cat hostlist`
```

выполнит команду `cat` в обратных кавычках, получит ее результат и подставит его обратно в команду, сформировав

аргументы для команды `ping`. Есть и альтернативная форма записи (которая мне нравится больше, так как мои дряхляющие глаза с трудом различают разные виды кавычек):

```
$ ping $(cat hostlist)
```

Подстановка команды часто используется в скриптах оболочки для присвоения вывода команды переменной в скрипте оболочки, например, так:

```
me=$(whoami)
```

64.25

Я не указал входной файл: как настоящий фильтр, Awk берет данные с клавиатуры, и я ввожу их вручную. Посмотрите внимательно, и вы увидите, что строки вывода чередуются со строками ввода. Для обработки каждого поля в программе используется классический цикл **for** в стиле языка C, в котором числа суммируются. Во встроенную переменную **NF** записывается количество полей в текущей строке ввода. Помните, что в этой программе шаблон (PATTERN) не задается, поэтому она выполняется над всеми входными строками.

Awk в реальном мире

Awk находит широкое применение в скриптах системного администрирования Linux, и большинство из них — всего лишь однострочные. Вот один такой, из файла **/etc/init/mounted-tmp.conf** в моей Ubuntu 12.04. Я сохранил окружающие строки, чтобы вы могли увидеть контекст:

```
# Проверим, хватит ли места в /tmp, и если нет, смонтируем tmpfs там
avail=`df -kP /tmp | awk 'NR==2 { print $4 }'`
if [ "$avail" -lt 1000 ]; then
mount -t tmpfs -o size=1048576,mode=1777 overflow /tmp
fi
```

В этом классическом примере Awk появляется в подстановке команды (см. врезку), чтобы задать значение переменной скрипта. Здесь Awk выбирает четвертое поле второй строки вывода команды **df**. Это не произвольный выбор — тот, кто писал скрипт, точно знал, каким будет вывод команды **df**.

Программы длиннее одной строки

Наши программы на Awk становятся длиннее, и набирать их в одной строке определенно все утомительнее; поэтому, прежде чем двигаться дальше, посмотрим, как поместить их во внешний файл скрипта. Это просто: поместите свои команды в файл и сошлитесь на него в командной строке с параметром **-f**. Вот программа, которая находит самый большой универсальный идентификатор в файле паролей:

```
BEGIN { maxuid = 0; FS = ":" }
{ if ($3 > maxuid) maxuid = $3 }
END { print "the largest UID is ", maxuid }
```

Если поместить эти строки в файл **maxuid**, то его можно будет запустить так:

```
$ awk -f maxuid /etc/passwd
```

Разберем эту программу. В ней три выражения. Специальный шаблон **BEGIN** применяется перед обработкой первой строки. Здесь мы используем его для инициализации парочки переменных, в том числе встроенной переменной **FS**, которая задает разделитель полей. Это альтернатива параметру **-F** в командной строке. Иногда шаблон **BEGIN** также используется для печати заголовков. У второго выражения нет шаблона, поэтому действие выполняется с каждой строкой. Это логика, отслеживающая самый большой UID (в третьем поле). Наконец, шаблон **END** означает, что мы обработали последнюю строку, и часто применяется для вывода окончательных результатов, как мы поступаем и здесь.

Переменная	Значение
ARGC, ARGV	Предоставляют доступ к аргументам командной строки, переданным Awk.
NF	Количество полей в текущей строке.
NR	Номер текущей записи (строки).
ENVIRON	Ассоциативный массив, предоставляющий доступ к переменным окружения программы.
FS	Разделитель полей входного потока (по умолчанию — пробел).
RS	Разделитель строк (по умолчанию — символ перехода на новую строку, но может использоваться регулярное выражение).
OFS	Разделитель полей выходного потока — используется для разделения полей, выводимых оператором print . По умолчанию — пробел.
IGNORECASE	Если параметр установлен, проверка регулярных выражений в Awk будет осуществляться без учета регистра.
FIELDWIDTHS	Разделенный запятыми список ширины полей для разделения входного потока на колонки фиксированной ширины.

Обратите внимание, что не нужно предварительно объявлять переменные или указывать их тип. Чтобы они появились на свет, достаточно лишь упомянуть их имя, а их типом станет тип присваиваемого им значения. Это сильно отличается от более традиционных языков программирования вроде C, в которых нужно объявлять все переменные и указывать их тип до их использования. Но такая «динамическая типизация» присуща более современным языкам программирования, наподобие PHP и Python.

Запустив эту программу, вы скорее всего увидите, что ответ равен 65534. Данный UID принадлежит несколько лживой учетной записи с именем "nobody". Чтобы игнорировать эту строку, можно добавить шаблон во второе выражение таким образом:

```
BEGIN { maxuid = 0; FS = ":" }
$1 != "nobody" { if ($3 > maxuid) maxuid = $3 }
END { print "the largest UID is ", maxuid }
```

Вот еще один пример, наугад взятый из Ubuntu, из файла **/etc/init.d/vmware**:

```
count=`/sbin/lsmmod | awk 'BEGIN {n = 0} {if ($1 == ""$driver"") n = $3} END {print n}'`
```

Это опять же пример подстановки команды, в котором переменная **\$driver** (определенная ранее в скрипте) довольно замысловато окружена кавычками.

Эмуляция других утилит

Awk — утилита общего назначения, которая умеет эмулировать всевозможные специализированные утилиты. Например, следующая команда добавляет номера строк в файл **foo** и эквивалентна **cat -n**:

```
$ awk '{ print NR, $0 }' foo
```

В этом примере подсчитывается количество слов во входном потоке, он эквивалентен **wc -w**:

```
$ awk '{ w += NF } END { print w }' foo
```

А этот пример по сути эквивалентен **grep**:

```
$ awk '/^chris/' /etc/passwd
chris:x:1000:1000:Chris Brown,,,:/home/chris:/bin/bash
```

➤ **Несколько встроенных переменных Awk.** Переменные, выделенные красным, предоставляют информацию, зеленым — управляют поведением.

История Awk

Awk был изначально написан в 1977 году и выпущен в Unix-версии 7 в 1978 году. В соответствии с духом нашего урока, его авторы писали: «Мы знали, как должен использоваться язык, и писали только

однострочные программы». Авторы языка расширили его в 1985 году, добавив среди прочего пользовательские функции. Их классическая книга «Язык программирования AWK» была написана в 1988 году.

Формально язык был определен в стандарте POSIX в 1992 году. Gawk (версия, которую вы найдете в Linux) появилась из проекта GNU, и в ней было добавлено много расширений.

Постойте — это еще не все!

В Awk есть масса встроенных функций. Есть математические функции, такие как `sin()`, `cos()`, `log()` и `sqrt()`, которые вам вряд ли пригодятся, и функции обработки строк, которые могут оказаться более полезными, от `length()`, просто возвращающей длину строки, до `split()`, разделяющей строку на фрагменты, и `gsub()`, выполняющей замену текста по регулярному выражению. Можно определить и собственные функции.

В этом примере у программы есть шаблон (регулярное выражение, соответствующее `^chris`), но нет действия. Действие Awk по умолчанию — просто вывести строку, как и делает `grep`.

Все следующие примеры основаны на «списке покупок», показанном на рисунке. Этот файл намеренно структурирован, и именно такие файлы любит разбирать Awk. Взгляните на этот файл, иначе вы не поймете примеры.

Сперва выведем общее количество товаров, которые мы наметили купить. Для этого достаточно суммировать значения во втором столбце:

```
$ awk '{ items += $2 } END { print items }' shopping
```

Затем отобразим суммарный счет на покупку. Для этого нужно умножить количество каждого товара на его цену и сложить результаты:

```
$ awk '{ cost += $2 * $4 } END { print cost }' shopping
```

Предположим, что нам нужно вычислить только стоимость инструментов («Хозтовары»). Для этого достаточно добавить регулярное выражение:

```
$ awk '/^Хозтовары/ { cost += $2 * $4 } END { print cost }' shopping
```

Теперь найдем самый дорогой товар:

```
$ awk '{ if ($4 > max) max = $4 } END { print max }' shopping
```

Здесь `max` — просто переменная, которую я придумал. Это не встроенная функция или что-то в этом роде. Опять же обратите внимание на динамическую типизацию — мне не нужно предварительно объявлять переменную `max` или инициализировать ее нулем.

Затем мы получаем список всех товаров (из файла «`lotsofthem`»), которых нам нужно по десять или больше штук.

➤ **Файл структурированного списка покупок, использованный в некотором примере.**

chris@m1530-1204: ~/Linux Format Articles			
Супермаркет	1	Курица	4.55
Супермаркет	50	Вешалки	1.25
Булочная	3	Хлеб	2.40
Хозтовары	1	Шланг	15.00
Одежда	1	Брюки	24.99
Хозтовары	2	Дверная ручка	8.40
Супермаркет	2	Молоко	1.25
Одежда	6	Носки	9.00
Хозтовары	2	Отвертка	2.00
Одежда	2	Юбка	28.00
Хозтовары	20	Наждак	10.00
Булочная	10	Плюшка	1.95
Булочная	2	Пирожок	6.50
Хозтовары	50	Гвозди	0.95
\$			

Этот пример нужен в основном для того, чтобы показать, как перенаправить вывод в файл:

```
$ awk ' $2 >= 10 { print $3 > "lotsofthem" }' shopping
```

Проясним, что символ «>» интерпретируется не оболочкой, а Awk. Он означает «записать вывод в указанный файл».

Вот действительно компактный пример, который разбивает данные на несколько файлов по одному для каждой категории:

```
$ awk '{ print > $1 }' shopping
```

Здесь \$1 (категория покупок) определяет имя выходного файла. После выполнения команды у вас появятся файлы **Булочная**, **Хозтовары** и т.д. Мне нравится этот пример! Если вас еще не впечатляет мощь однострочных программ, я сдаюсь.

Следующий пример разбивает расходы по категориям. Здесь я поместил программу Awk во внешний файл под названием `catcost`:

```
{ cost[$1] += $2 * $4 }
END { for (cat in cost) print cat, cost[cat] }
```

...и запустил программу так:

```
$ awk -f catcost shopping
```

```
Булочная 39.7
Одежда 134.99
Хозтовары 283.3
Супермаркет 69.55
```

В этом примере используется ассоциативный массив (массив, индекс += \$2 * \$4)

```
3. END {
4.   max = 0;
5.   for (cat in cost) {
6.     if (cost[cat] > max) {
7.       max = cost[cat];
8.       maxcat = cat;
9.     }
10.  }
11.  print maxcat;
12. }
```

Нужно некоторое пояснение. Строка 1 — комментарий. Во второй строке, как в предыдущем примере, стоимость по категориям заносится в ассоциативный массив. Строки 3–12 являются частью действия END. Обратите внимание, что теперь это больше похоже на обычную программу, и мы видим, что выражения завершаются символом ;. В строках 5–10 мы пробегаемся в цикле по категориям, определяя наибольшую стоимость. Если вы уже запыхались, можете облегченно перевести дух — это самая длинная программа на Awk, которую я показал!

Самодостаточные скрипты

Также можно писать самодостаточные скрипты Awk и запускать их прямо из оболочки. Это немного удобнее, чем каждый раз набирать `awk -f` в командной строке. Вот как это сделать.

Во-первых, добавьте «шапку» в начало скрипта, чтобы Linux знал, какой интерпретатор использовать, например, так:

```
#!/usr/bin/awk -f
{ cost[$1] += $2 * $4 }
END { for (cat in cost) print cat, cost[cat] }
```

Затем сделайте скрипт исполняемым, как и любой другой скрипт:

```
$ chmod u+x catscript
```

Теперь можно запустить скрипт напрямую:

```
$ ./catscript shopping
```

Если вам нравятся однострочные программы, загляните в www.pement.org/awk/awk1line.txt. Если хотите увидеть более длинные программы на Awk, загрузите руководство по gawk по ссылке www.gnu.org/software/gawk/manual. Классическая книга по языку — «Язык программирования Awk» Ахо, Вайнбергера и Кернигана. Ее стоит поискать на eBay ради впечатляющей ясности, с которой она написана. Пока! **LXF**

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Игры и Linux

— Игра на всю жизнь!
— Какие игры в такую рань?!
*Каникулы Петрова и Васечкина,
обыкновенные и невероятные*

Все, теперь уже нельзя говорить, что в GNU/Linux нет игр. Вышел широко известный в узких кругах *Football Manager 2014*, также и для GNU/Linux. Появились *Metro: Last Light* и *Shadowrun Returns* в Steam для Linux. Конечно, не любая игра сейчас сразу доступна под нашей платформой, но тенденция Linux-портов за последние два года стала неоспоримым фактом. Если нет тяги к конкретному бренду, то найти, во что поиграть, уже не проблема. Сейчас можно действительно выбрать, а не хватать все, что дают.

Но... грядущее игровое изобилие особо не влияет на рост популярности GNU/Linux как платформы. И уж точно не влияет на развитие свободного игрового (и около) контента. Бизнес повернулся в сторону GNU/Linux не ради поддержки платформы и сообщества: просто наше сообщество уже не проигнорировать. Жить стало комфортней, но развитие сообщества и платформы пока в основном определяется сообществом; что, возможно, и к лучшему.

PS Проснулись патентные тролли из Rockstar Bridco. Грядут интересные времена для Android и Google. Скорее всего, «бэбл» победит, но как бы осколками нас не задело.

PPS Развивается альтернатива официальному магазину приложений Android — пакетный менеджер и репозиторий F-droid; в нем программы (только свободные) собираются из исходников и подписываются.

e.m.baldin@inp.nsk.su

В этом месяце вы научитесь...



Болтать через IRC 62

Джозо Бэкон пользуется старым добрым Internet Relay Chat, чтобы находить друзей и влиять на совершенно посторонних личностей. И вот как это делается...



Делиться файлами 66

Узнайте от **Нейла Ботвика** всю подноготную файловых систем, наиболее удобнейших для обмена файлами с вашими замшелыми дружками из Виндовилля.



Делать резервные копии 68

Освойте Open Media Vault, установите свое решение NAS — и вы больше не потеряете ни одного файла, как и всегда-готовый **Маяк Шарма**.



Перестраивать ядро 72

Наберитесь внутренней мудрости (подобно **Майку Сондерсу**), раздербанив потроха вашей операционной системы. Если системы не жалко.



Собирать приложения ... 76

Изучите с **Джозо Бэконом**, как воспользоваться SDK Ubuntu при построении приложений для тачскринов в мобильной ОС Ubuntu.



Вникать в Erlang 80

Андрей Ушаков завершает свой сериал по многозадачности, оценивая производительность функций. Портит картину только мусорщик.



Следить за движением 84

В любимом *Blender*'е **Андрею Прахову** удается даже превращать людей в роботов. Но не навеки, не бойтесь!



Гулять по дорожкам 88

Включая в *LiVES* режим Multitrack, **Сергей Яремчук** рекомендует быть осторожным. Стабильность приходит с опытом.



Отбивать web-атаки 92

Игорю Штомпелю нечего опасаться кибер-угроз: на страже стоит *Linux Malware Detect*. Последуйте хорошему примеру и тоже его установите.

IRC: Давайте

Джоно Бэкон берет нас на экскурсию по IRC, одному из популярнейших мест тусовки любителей открытого ПО.



Наш эксперт

Джоно Бэкон управляет сообществом Ubuntu, он автор книги «Искусство сообщества» и основатель ежегодного Саммита лидеров сообщества.



Годами люди раздумывали о важных ингредиентах, сыгравших свою роль в глобальном успехе Linux и движения открытого ПО. Некоторые предположили, что успех вызван доступностью высококачественных утилит (таких как GCC — набор компиляторов GNU, Make от GNU и X Window System) или, возможно, публичной лицензией GNU, а другие объясняли его

удешевлением доступа в Интернет и поддержкой Linux в университетах. Теорий, короче, хватает. Хотя все эти вещи определенно сыграли важнейшую роль, я считаю, что свой вклад внес еще один ингредиент, предоставивший основу, на которой многие открытые проекты могли планировать свое развитие и развиваться, а также создавать гостеприимные и тесные сообщества — это IRC (Internet Relay Chat — ретранслируемый интернет-чат). На нашем уроке мы исследуем эту стойкую коммуникационную среду: посмотрим, откуда она взялась, как ею пользоваться и каковы социальные нормы, принятые в IRC-сообществах.

Немного истории

Ретранслируемый интернет-чат, известный большинству людей как IRC — текстовая среда обмена информацией, которая используется во всем мире, охватывает несколько стран, сотни проектов и тысячи пользователей. Финн Яркко Ойкаринен [Jarkko Oikarinen] создал IRC еще в горячие дни 1988 года. Яркко был активным пользователем финского BBS-форума *OuluBox*. Он устал от программы *MUT* — сокращение от “Multi User Talk [Многопользовательский диалог]” — которую он применял для общения с другими пользователями. Вдохновленный другим чатом, *Bitnet Relay* (от него произошли многие команды современного IRC), Яркко написал самый первый IRC-чат.

Базовые принципы IRC не слишком изменились с тех далеких дней, когда Яркко создал свою первую версию. Давайте кратко пробежимся по тому, как он работает.

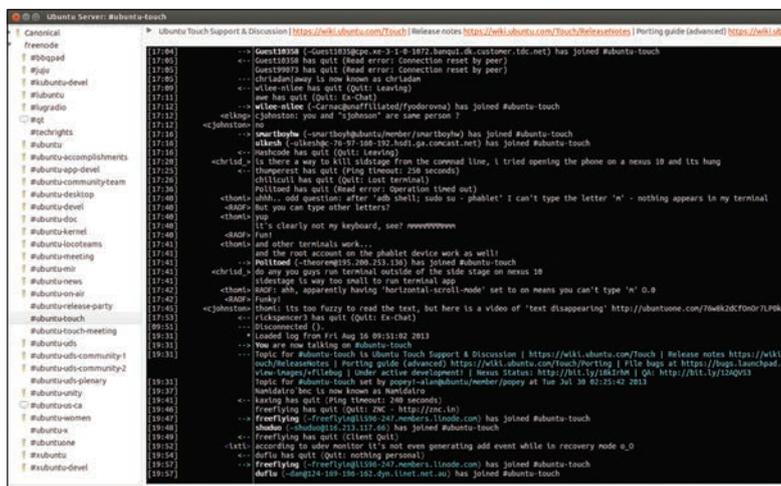
В центре всего находится IRC-сервер. Это компьютер, к которому вы подключаетесь специальной программой, называемой IRC-клиентом. Такие программы знают, как сладко-сладко уболтать сервер, и сервер возвращает их слова соответствующим клиентам. Серверы расщеплены по всему миру, подавляющее большинство их бесплатно, и они привлекают желающих пообщаться со всего мира, создавая по-настоящему глобальную аудиторию.

При подключении к серверу IRC нужно выбрать имя пользователя (оно же часто — просто «ник»). Это короткая строка, которая идентифицирует вас. Это не профиль, где можно указать свой сайт, поставить аватаром своего кота и кратко описать свою биографию. Ник IRC — это буквально ссылка на вас в Сети, и когда вы выходите из чата, то же происходит и с ником.

После ввода ника можно присоединиться к IRC-каналу. Канал — это виртуальная комната, к которой также подключены другие пользователи. У большинства каналов есть какая-то тема или общее назначение. Например, канал #linuxformat может быть посвящен обсуждению журнала, а канал #openstreetmap — обсуждению проекта OpenStreetMap.

Подключившись к каналу, можно набрать текст, который увидят другие члены канала. В подавляющем большинстве IRC-клиентов ваш ник отображается в квадратных скобках рядом с текстом, который вы набираете. Например:

- Jonno:** всем доброе утро! как дела?
- Graham:** отлично, спасибо!
- Sarah:** очень хорошо, спасибо, Джоно!



» Мой клиент *XChat-Gnome* в действии, с набором каналов, в которых я обычно провожу большую часть своего времени в Сети.

ПОДКЛЮЧИМСЯ

Здесь вы видите, как я задал вопрос, а двое других пользователей ответили на него.

Важно отметить, что когда вы что-то пишете в канал IRC, все остальные в этом канале могут это прочесть. Например, пользователь bob может активно не переписываться с вами или с другими пользователями, но все равно увидит разговор, который произошел выше. Конечно, для этого пользователь bob должен находиться в канале; если Боба в канале нет, разговора ему не видать.

Чем же IRC так удобен? Когда кто-то набрал текст и нажал клавишу Enter, набранный им текст появляется в канале. Это делает IRC средой общения реального времени, то есть, когда вы разговариваете с другим пользователем, все выглядит более непосредственным и подлинным по сравнению с долгими и затянутыми диалогами в других средах обмена информацией, например, в списках рассылки.

Хотя большая часть дискуссий на IRC-серверах происходит в этих IRC-каналах, можно общаться с другим пользователем и приватно, один на один.

Это совсем другой тип канала, и он называется приватными сообщениями (некоторые пользователи часто сокращают это название до "privmsg" или "msg"). Важное различие

между IRC-каналами и разговорами один на один в том, что в последнем случае разговор является приватным, и другие пользователи не видят, что вы набираете. Но помните, что вы все равно набираете текст, который можно скопировать и вставить, поэтому не делитесь своими самыми глубинными тайнами и темными секретами с кем-то, кто может скопировать и вставить их и поделиться ими с другими, не уведомляя вас об этом.

Подключаемся

Получив представление о том, как работает IRC, подключимся к IRC-серверу и начнем общаться с незнакомцами! Для подключения нам понадобится программа, специально предназначенная для преобразования «бормотания» IRC-сервера в формат, доступный вам. Она называется IRC-клиентом, и существует

множество клиентов для различных рабочих столов, устройств и платформ. Для полноты рассмотрим один клиент для рабочего стола, один для командной строки и один с web-интерфейсом. Чтобы немного повеселиться, мы также расскажем о некоторых клиентах для мобильных устройств (см. врезку «IRC на мобильных» на стр. 64).

Клиенты IRC для рабочего стола

Существует масса различных IRC-клиентов для рабочего стола. Наверное, самый популярный из них — *XChat-Gnome* (<https://launchpad.net/xchat-gnome>), мощный клиент на базе устаревшего кода *XChat*, адаптированного для рабочего стола Gnome и его производных. Можно упомянуть и *Konversation* (<http://konversation.kde.org>), популярный клиент для KDE.

Мы не будем описывать их установку в вашем рабочем столе — просто установите последнюю версию пакета для своего дистрибутива в своем любимом центре программ или менеджере пакетов. Но как начать работу в одном из клиентов, *XChat-Gnome*, мы покажем.

Прежде чем мы к этому приступим, запомните, что между *XChat* и *XChat-Gnome* есть разница. Хотя *XChat* вполне годится и сейчас, настоятельно рекомендую установить *XChat-Gnome*, пользовательский интерфейс и настройки по умолчанию которого гораздо лучше, чем у *XChat*.

Установив *Xchat-Gnome*, запустите его. Теперь выберите IRC > Connect [IRC > Подключение], и вы увидите список сетей на выбор. Дважды щелкните на одной из них, и соединение будет установлено. После этого вам предложат список каналов. Например, дважды щелкните на сети Freenode (самая популярная сеть для проектов открытого ПО) и зайдите в канал #ubuntu или #fedora, чтобы увидеть их в действии.

С левой стороны окна вы видите все сети, к которым вы подключены (при желании можно подключиться к нескольким сетям одновременно), а в основной части окна отображается текущий разговор IRC. Для отправки текста в канал введите его в поле в нижней части окна.

«Не делитесь своими глубинными тайнами и темными секретами.»

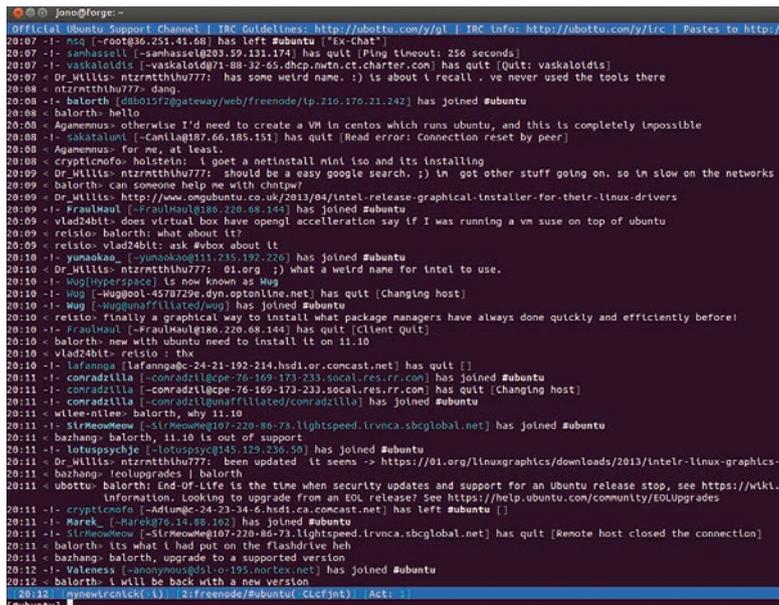
Нормы общения

Как и в любой коммуникационной среде, в IRC есть набор норм общения. Вот несколько из них:

- » Избегайте расистских, сексистских, оскорбительных высказываний и ругательств как в публичном канале, так и в приватном чате.
- » Не наводняйте каналы ссылками и неконкретной информацией.
- » Старайтесь быть дружелюбным и узнавать других людей в канале.

- » Не рекламируйте другие каналы в том, в котором находитесь.
- » Если кто-то просит вас соблюдать правила, не пытайтесь обороняться или проявлять агрессию в ответ на эту просьбу.
- » Не набирайте сообщения в канале ЗАГЛАВНЫМИ БУКВАМИ (они интерпретируются как крик).
- » Не копируйте в канал большие фрагменты текста.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



Ирssi в терминале, подключенный к одному каналу.

Хороший совет, пригодный в большинстве клиентов — пользоваться автодополнением имени пользователя, набрав первые несколько букв и нажав Tab. Например, для ника flibbidygibbits можно набрать “fl” и нажать Tab, и имя будет дополнено.

Кстати, когда кто-то в IRC-канале указывает ваше имя, оно выделяется желтым цветом, чтобы легко отделять сообщения, относящиеся к вам, от остального чата. Это по-настоящему удобный способ увидеть, что обращаются к вам (или упоминают вас), когда в канале много других разговоров.

Настройка автоподключения

Во многих случаях вы захотите увидеть, какие пользователи в данный момент зашли в канал. Для этого щелкните по тексту “XX Users” (XX — число) под списком каналов, и список пользователей откроется в отдельном окне. Если пользователь выделен серым, это означает, что он установил статус «отшел» (подробнее об этом позже).

Щелкнув на пользователе в списке левой кнопкой мыши, можно начать с ним приватный разговор, а правой — увидеть другие функции (многие из них будут интересны операторам каналов, но о роли оператора мы подробнее поговорим позже).

По мере обнаружения новых каналов и сетей вы, возможно, захотите подключаться к ним автоматически при запуске XChat-Gnome. Сделать это легко: щелкните правой кнопкой мыши

на сети и выберите “Auto-connect on startup [Автоматически подключаться при запуске]”. Тогда клиент автоматически подключится к сети; а чтобы подключиться к своим любимым каналам этой сети, щелкните правой кнопкой на канале и выберите “Auto-join on startup [Автоматически заходить в канал при запуске]”. Повторите эту процедуру для всех необходимых сетей и каналов. Например, в своем клиенте XChat-Gnome я подключен к двум сетям с 12 каналами в одной и с 24 в другой. Это все мои IRC-потребности в одной посуде.

Подключение с консольным клиентом

В мире IRC выбор консольного клиента — тема для жаростных дискуссий, но в большинстве случаев на ум приходит долгожитель Irssi. Irssi содержит большой набор возможностей и функциональность, которые хотя и сложнее, чем у среднего настольного клиента, но легко осваиваются, если приложить немного усилий.

К тому же, научившись работать с Irssi, вы сможете заходить в IRC с любого терминала, будь то ваш компьютер или SSH-подключение. Поэтому именно Irssi выбирают системные администраторы во всем мире.

Установив программу из любимого центра программ или менеджера пакетов, запустите ее, набрав Irssi в терминале. После этого вы увидите две синие строки, а под нижней синей строкой находится строка, где можно набирать сообщения. В этих синих строках происходят обновления статуса. В строке внизу можно набирать текст, а также специальные команды IRC, и вводить команды для доступа к различным функциям сети IRC и самой программы.

Начнем с подключения к сети. Подключитесь к популярной сети Freenode, набрав `/connect irc.freenode.net`. В Irssi начнут появляться сообщения о подключении. Они появляются в окне состояния клиента. Чтобы войти в канал, наберите: `/connect #имяканала`, например, `/connect #ubuntu`.

Повторите эту команду для всех каналов, куда хотите зайти. Заходя в очередной канал, вы заметите, что предыдущий канал исчез. К счастью, это не так: следующий канал просто появляется поверх него. Для переключения между каналами пользуйтесь клавишами Alt+2, Alt+3 и т.д. По нажатию клавиши Alt+1 открывается окно состояния, о котором мы писали выше.

Наш совет для XChat-Gnome, насчет автодополнения имени пользователя по клавише Tab, работает и в Irssi. Например, если ник пользователя “flibbidygibbits”, можно набрать “fl” и нажать Tab, чтобы дополнить остальное.

Как и в предыдущем клиенте, когда кто-то в канале пишет ваше имя, оно выделяется желтым цветом.

Использованию и настройке Irssi можно посвятить отдельную статью (подробнее читайте на сайте <http://irssi.org/documentation>), но есть еще одна хорошая возможность, о которой я хочу рассказать, прежде чем двигаться дальше: разбиение окон.

В Irssi можно разбить окно на части и наблюдать за несколькими чатами одновременно. Для создания нового окна просто наберите `/window new`.

После создания нового окна ему назначается новое сочетание клавиш Alt+XX. Например, у вас может быть такая схема:

- » Alt+1 — `/connect irc.freenode.net`
- » Alt+2 — `/join #ubuntu`
- » Alt+3 — `/window new`
- » Alt+4 — `/join #fedora`

«Именно Irssi выбирают системные администраторы во всем мире.»

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Исчерпав клавиши с Alt+1 по Alt+0, можно перейти к сочетаниям с Alt+Q до Alt+P. Это предоставляет массу способов подключиться к различным каналам и сетям и работать с ними.

Хотя для подключения к IRC можно воспользоваться клиентом рабочего стола или терминальным клиентом, иногда может возникнуть необходимость подключиться оттуда, где вы не можете установить IRC-клиент (например... кхе-кхе... на работе). В этих случаях удобно воспользоваться браузером.

Сегодня большинство IRC-сетей используют собственные web-интерфейсы, а также используют для этого один и тот же web-клиент *Qwebirc*. Например, чтобы подключиться к популярной IRC-сети Freenode, зайдите на <http://webchat.freenode.net>.

Подключение через web-клиент

Подключиться легко: введите ник и канал, заполните CAPTCHA, нажмите кнопку Connect [Подключиться] — и все. После подключения этот клиент очень похож на настольный: в основной части окна вы видите чат, справа — список ников, а внизу — строку для ввода сообщения.

Для входа в новые каналы набирайте `/join #имяканала`, и каждый канал будет открываться в новой вкладке. Наконец, есть несколько настроек, задаваемых щелчком по иконке в левой верхней части страницы и выбором Options [Параметры].

Теперь, когда мы рассмотрели основные доступные клиенты, пробежимся по общим функциям IRC, которые понадобятся вам в этом бесстрашном рейде в альтернативную среду.

У каждого IRC-сервера есть общий набор команд, которые полезно знать, если вы намерены пользоваться IRC каждый день. Все эти команды поддерживаются всеми клиентами. Самая важная команда, как мы уже говорили ранее — команда входа в канал `/join`, а ее антиподом является команда выхода из канала — `/part`.

Еще одна полезная команда предназначена для показа, что вы все еще подключены, но отошли от компьютера. Это `/away`.

Во многих клиентах это означает, что вы отошли от компьютера, и программа часто сообщает об этом другим пользователям, которые пытаются отправить вам сообщение. Также можно указать причину: `/away Ушел к зубному`. По возвращении можно отключить режим `/away` командой `/back`.

Наконец, вы можете захотеть изменить свой ник. Для этого достаточно выполнить команду `/nick новыйник`. Она установит ваше имя в “новыйник”. Некоторые IRC-сети содержат специальный набор сервисов для отслеживания и резервирования ников. Эти сервисы различаются, поэтому доподлинно о них лучше узнавать в официальной документации для каждой сети.

Создание и модерация каналов

Возможно, вам все время было интересно, как появились все эти каналы и можно ли создать собственный. К счастью, в подавляющем большинстве сетей создать канал может любой желающий.

Для этого достаточно зайти в канал, которого еще не существует. Заходя в этот канал, вы по определению создаете его, и другие могут свободно присоединиться к вам в канале. Например, чтобы создать канал `#flibbidygibbits`, нужно набрать `/join #flibbidygibbits`. Бум! Чат `flibbidygibbits`, о котором вы всегда мечтали, готов!

При создании канала вы становитесь его оператором (часто сокращается до “op”) и можете делать других пользователей операторами каналов, выгонять их из чата и изгонять их навеки (желательно только тогда, когда они действительно неправы, иначе вам грозит одиночество в пустом канале). Эти команды

```
#ubuntu - freenode Web IRC
webchat.freenode.net
Status: #ubuntu
[Official Ubuntu Support Channel | IRC Guidelines: http://ubuntu.com/irc | IRC info: http://ubuntu.com/irc | Pastes to http://paste.ubuntu.com/ | Release Notes: http://ubuntu.com/
Supported versions 10.04 LTS (server), 12.04 LTS, 12.10 and 12.04]
[20:07] == samhassel1 [-samhassel@203.59.131.174] has quit (Ping timeout: 256 seconds)
[20:07] == vaskaloidis [-vaskaloidis@71-88-32-65.dhcp.metn.ct.charter.com] has quit [Quit: vaskaloidis]
[20:08] <ntzrntthhu777> dang.
[20:08] == balorhn [d88015f2@gateway/web/freenode/sp.216.176.21.242] has joined #ubuntu
[20:08] <balorhn> hello
[20:08] <Agamenus> otherwise I'd need to create a VM in centos which runs ubuntu, and this is completely impossible
[20:08] == sakatatum [-Camila@187.66.185.151] has quit (Read error: Connection reset by peer)
[20:08] <Agamenus> for me, at least.
[20:08] <crypticmofo> holstein: i got a netinstall mini iso and its installing
[20:09] <Dr Willis> ntrzrntthhu777: should be a easy google search. ;) im got other stuff going on. so im slow on the netw
[20:09] <balorhn> can someone help me with chnpt?
[20:09] <Dr Willis> http://www.omgubuntu.co.uk/2013/04/intel-release-graphics-installer-for-their-linux-drivers
[20:09] == FrauHaut [-FrauHaut@186.220.68.144] has joined #ubuntu
[20:09] <vlad24bit> does virtual box have opengl acceleration say if I was running a vm suse on top of ubuntu
[20:09] <reissio> vlad24bit: ask avbox about it
[20:10] == yumaokao [-yumaokao@111.235.192.226] has joined #ubuntu
[20:10] <Dr Willis> ntrzrntthhu777: 0i.org :) what a weird name for intel to use.
[20:10] == WugjHyperspace has changed nick to Wug
[20:10] == Wug [-Wug@001.45787296.dyn.optonline.net] has quit [Changing host]
[20:10] == Wug [-Wug@unaffiliated/wug] has joined #ubuntu
[20:10] <reissio> finally a graphical way to install what package managers have always done quickly and efficiently before!
[20:10] == FrauHaut [-FrauHaut@186.220.68.144] has quit (Client Quit)
[20:10] <balorhn> new with ubuntu need to install it on 11.10
[20:10] <vlad24bit> reissio : thx
[20:10] == lafanna [lafanna@24-21-192-214.hsdl.or.comcast.net] has quit []
[20:11] == comradzilla [-comradzill@cpe-76-169-173-233.socal.res.rr.com] has joined #ubuntu
[20:11] == comradzilla [-comradzill@cpe-76-169-173-233.socal.res.rr.com] has quit [Changing host]
[20:11] == comradzilla [-comradzill@unaffiliated/comradzilla] has joined #ubuntu
[20:11] <uilee-nilee> balorhn, why 11.10
[20:11] == SirMeowMeow [-SirMeowMeow@107-220-86-73.lightspeed.irvnc.sbcgloab.net] has joined #ubuntu
[20:11] <bazhang> balorhn, 11.10 is out of support
[20:11] == lotuspysnje [-lotuspysnje@145.120.236.50] has joined #ubuntu
[20:11] <Dr Willis> ntrzrntthhu777: been updated it seems -> https://01.org/linuxgraphics/downloads/2013/intel-linux-graphics-version-1.0.2
[20:11] <bazhang> leolupgrades | balorhn
[20:11] <dotu> balorhn: End-of-Life is the time when security updates and support for an Ubuntu release stop, see https://
for more information. Looking to upgrade from an EOL release? See https://help.ubuntu.com/community/EOLUpgrades
[20:11] == crypticmofo [-Adium@c-24-23-34-6.hsdl.ca.comcast.net] has left #ubuntu []
[20:11] == Marek [-Marek@76.14.88.162] has joined #ubuntu
[20:11] == SirMeowMeow [-SirMeowMeow@107-220-86-73.lightspeed.irvnc.sbcgloab.net] has quit [Remote host closed the connectio
```

➤ Web-клиент IRC в Firefox, позволяющий получить доступ к IRC из любого места.

довольно просты. Чтобы сделать пользователя оператором канала, используется команда `/op ник`. Например, команда `/op bob` сделает оператором Боба. Выгнать и насовсем изгнать пользователей тоже просто, Команда `/kick ник “причина”` (например, `/kick dave “хватит ругаться”`) выкинет пользователя из канала, но затем он снова сможет зайти в канал. После команды `/ban ник` пользователь удаляется навсегда, то есть в канал ему будет уже не вернуться.

Недостаток такого подхода к созданию каналов в том, что вы можете уйти из канала, а канал затем создаст кто-то другой, у кого есть привилегии оператора. К счастью, во многих IRC-сетях можно зарегистрировать канал, чтобы вы могли выйти из канала и все равно остаться его оператором. Для этого используется

команда `/msg chanserv register #мойканал` (для сетей, которые это поддерживают; опять же, ознакомьтесь с официальной документацией). На этом все настройки канала закончены!

В нашем начальном руководстве мы кратко познакомились с IRC: узнали что это, как к нему подключиться и как им пользоваться. IRC — по-настоящему чудесная среда, подключиться к которой точно стоит; а если у вас есть любые вопросы, то в IRC есть масса людей, которые помогут вам продолжить свое приключение. Удачи! LXF

«IRC — чудесная среда, подключиться к которой точно стоит.»

Встраивание IRC

Одно из самых замечательных преимуществ клиента *Qwebirc* — в том, что он используется во многих сетях IRC, и это означает, что его легко встроить в свой собственный сайт.

Например, если я хочу встроить в свой сайт виджет Freenode и открыть на нем канал `#мойканал`, можно встроить в сайт следующий код:

```
<iframe src="http://webchat.freenode.net?channels=%23mychannel&uio=d4" width="647" height="400"></iframe>
```

Для изменения размера фрейма можно использовать обычные атрибуты тэга `<iframe>`. Эта возможность удобна для сайтов сообществ, если вы хотите, чтобы пользователи легко зашли в ваш IRC-канал с сайта.

Различные ФС:

FAT, NTFS, FUSE и exFAT... **Нейл Ботвик** объясняет все за и против каждой из них для обмена информации с Windows.



Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не расскажет, где находится центральный сервер.

Как бы мы ни любили настольный Linux, следует признать, что пользователей Windows куда больше, и периодически надо обмениваться с ними информацией. Хотя сервисы обмена файлами, такие как Dropbox и ownCloud, и даже старая добрая электронная почта значительно облегчают электронный обмен информацией, иногда ничто не заменит старого доброго переноса данных вручную с одной машины на другую.

Будь то быстрый обмен парой фотографий на USB-брелке или сотни гигабайт на внешнем жестком диске, нам нужен носитель, который смогут читать и на который смогут делать запись пользователи всех платформ. Я недавно слышал жалобу, что хотя в *Linux Format* много пишут про Linux, но никогда не упоминают о форматировании, и в этом месяце мы попробуем исправиться, рассмотрим разные опции форматирования в разных файловых системах.

FAT: мощная, но мелкая

[игра слов: fat — толстый, жирный, — прим. пер.] Фактическим стандартом для взаимозаменяемых файловых систем (ФС) долгое время была файловая система Windows FAT. Она отлично служила нам при переносе данных посредством USB-брелка, и до сих пор большинство флэш-дисков форматированы именно так, когда вы их приобретаете, но у нее есть свои ограничения. FAT — 32-битная система, и максимальный размер файлов в ней — 4 ГБ; вот вам еще одно проявление синдрома «640к должно хватить любому», когда на момент создания ограничение казалось более чем мягким. Это основное ограничение; его можно обойти посредством сжатия или нарезки файлов, но файлам нельзя превышать 4 ГБ.

Другая проблема, влияющая на применение FAT в Linux — это проблема разрешений. FAT не имеет поддержки имени владельца и прав доступа к файлам, принятых в Linux. При монтировании файловой системы к ней применяются своего рода псевдо-метаданные. Это происходит по умолчанию со всем, на что распространяются права пользователя, монтирующего ее, и только этот пользователь может делать в ней записи. Когда вы используете для этой задачи автоматизированный рабочий стол, оно и неплохо, поскольку все монтируется так, чтобы собственником был пользователь, работающий в системе. Если вы монтируете из `/etc/`

`fstab` или из терминала — как суперпользователь-root или через `sudo` — только root сможет делать запись на устройство. Одним из решений будет использовать `pmount`, который монтирует съемное устройство как пользователь, сохраняя права собственности.

Другое решение, которое тоже работает, если вам нужно, чтобы доступ к устройству имел не один пользователь — изменить настройку `umask` при монтировании. Следующий код подмонтирует устройство с полными разрешениями `gwx` для всего:

```
mount /dev/sdb1 /mnt/usbstick -o umask=000
```

Более безопасная опция —

```
mount /dev/sdb1 /mnt/usbstick -o dmask=000,fmask=111
```

Она настроит директории на `gwx`, а файлы — на `gw` (буква `x` здесь означает, что файл или директория исполняемые). Вы также можете настроить права собственности на файловую систему с помощью опций `uid` и `gid`. Все эти опции даются в командной строке, как здесь, или в `/etc/fstab`.

NTFS

NTFS (New Technology Filesystem) от Microsoft разработана как замена FAT, и вот уже много лет используется на жестких дисках. Однако поддержка NTFS в ядре Linux никогда не была сильна. Даже в самом свежем ядре (3.10.9), NTFS, за редким исключением, надежно работает только для операций чтения.

Файлы в NTFS могут безопасно записываться, только если файл уже существует и новый файл имеет ту же длину, что и старый. Другими словами, вы можете менять содержимое коробки, но не ее размер. Это очень жесткое ограничение, но даже такая возможность существует, только будучи специально включенной в ядре, чего многие дистрибутивы не делают.

Однако есть файловая система FUSE (см. врезку внизу *FUSE: Мир файловых систем*), которая дает полную поддержку чтения/записи для NTFS под названием NTFS-3G. Она должна быть включена во все дистрибутивы и работает так же, как любая другая файловая система, пока подгружается модуль ядра `fuse`. Это драйвер NTFS по умолчанию для большинства дистрибутивов; вы должны быть в состоянии подключить устройство, форматированное в NTFS, и либо разрешить рабочему столу подмонтировать его для вас, либо подмонтировать его из командной строки, как

FUSE: Мир файловых систем

Linux реализует файловые системы внутри ядра, так же, как большинство других драйверов оборудования. Это может быть в дереве кода, как обычно и случается, или в стороннем модуле, как это происходит с ZFS. Ну, а если у вас файловая система, которая не может или не будет соединяться с ядром?

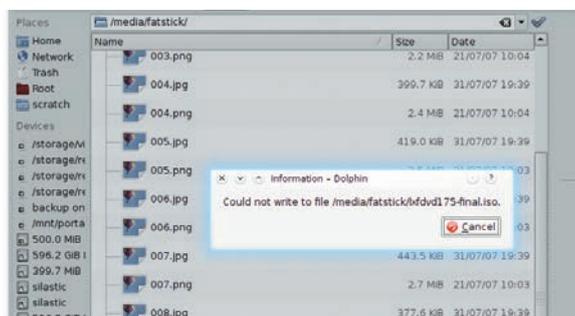
FUSE (Filesystem in Userspace — файловая система в пространстве пользователя) предоставляет решение — способ создать драйверы файловых систем, не соединяя их с ядром ссылкой. Вся магия

ядра вершится в модуле FUSE, а драйвер говорит с ним из пространства пользователя. Что касается ядра, файловая система любого устройства, подмонтированная через FUSE, это Fuseblk. Неважно, как форматировалось само устройство: команды типа `mount` или `df -T` покажут тип файловой системы как Fuseblk. Таковы традиционные файловые системы, реализованные через FUSE, но есть и другие модули, представляющие информацию в виде файловой системы. Если у вас есть SSH-доступ к удаленной системе, но нет на ней файловой системы

с распределенным сетевым доступом, можете использовать SSHFS, чтобы подмонтировать ее.

Хотя ее вытесняет Encrptfs ядра, файловая система FUSE, EncFS, может хранить содержимое директорий в зашифрованном виде. Еще меньше похожа на файловую систему MP3FS, которая перекодировывает аудиофайлы FLAC в MP3 на лету, так что вы можете хранить свою музыку как FLAC, но копировать ее в свой плеер, как MP3. И это, и еще много другое можно найти на SourceForge: <http://bit.ly/1adSgm0>.

Делим файлы



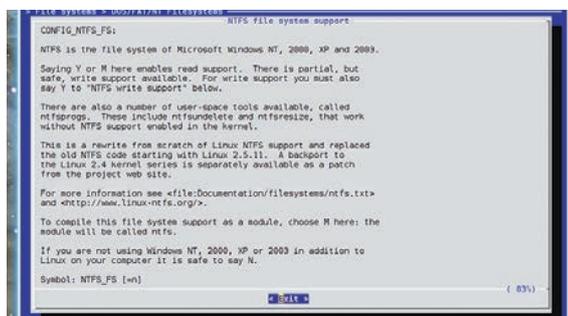
➤ FAT работает с мелкими файлами, не больше фотографий, но не справляется с обработкой крупных.

любую другую ФС. Хотя разрешения файлов NTFS по-прежнему несовместимы с метаданными Linux, NTFS-3G настроен по умолчанию на `umask` с 0, поэтому все полностью читается, пишется, и разрешения исполняются. Если вы не хотите, чтобы у файлов был установлен бит исполнения, можно назначить опции `dmask` и `fmask`, как для FAT. Но для директорий бит исполнения установить придется, иначе будет не прочесть их содержимое.

FAT набирает вес

NTFS предназначена для использования на жестких дисках и изначально разрабатывалась для Windows NT. Большинство ее функций предназначены именно для дисков, а не для небольших по объему съемных устройств хранения данных. FAT прекрасно работала, пока не стала слишком мала для всего. Например, когда камеры просто делали фотографии и хранили их на картах памяти объемом менее 4 Гб, ограничения на размер не были проблемой. Теперь камеры записывают видео, а 64-Гб SD-карты стали дешевле, чем я когда-то заплатил за 16-Мб CF-карту, и данное ограничение уже неприемлемо.

Ответом Microsoft стала exFAT, файловая система для флэш-накопителей без ограничений FAT или затрат NTFS. Это ФС по умолчанию для карт SDXC. exFAT имеет приличную коллекцию патентных и лицензионных соглашений, вполне достаточную для того, чтобы ее включение в ядро стало проблематичным, поэтому свободная опция использует FUSE. Ситуация с лицензированием делает серой областью даже включение



➤ Страницу подсказки ядра по встроенной поддержке NTFS не назовешь вдохновляющим чтением.

версии FUSE в дистрибутив. А значит, пакет exFAT редко устанавливается по умолчанию; но обычно он доступен в репозиториях по умолчанию менеджера пакетов. Там обычно имеется два пакета для установки, `exfat-fuse` и `exfat-utils`. Первый — драйвер ФС и обработчик монтирования, и этого вам хватит для монтирования существующих файловых систем exFAT. Второй содержит всякие утилиты и нужен, чтобы создавать, ремонтировать, переразмечать или обследовать файловые системы exFAT. Установив пакет, можно подмонтировать и работать с устройствами, форматированными в exFAT, как с любой другой ФС.

Уж лучше вы к нам?

Пока мы изучали возможности работы с файловыми системами Windows в Linux; а если наоборот? И как насчет Mac OS X? С OS X все просто: она умеет читать большинство файловых систем Windows. Работа с файловыми системами Linux в любой из них возможна, но ограничена. На SourceForge есть драйверы для ext2 и для Windows, и для MacOS (драйвер Mac использует FUSE), но они не обеспечивают надежной работы с ext4, которая долго была стандартной ФС для большинства дистрибутивов Linux; так что, как правило, безопаснее всего будет заставить Linux работать с ними, а не пытаться убедить их сделать что-то по нашему хотению. Вам в любом случае не понадобится ext3 или ext4 на флэш-накопителе USB, поскольку журналирование истреплет его быстрее (то же относится и к NTFS), так что можете с тем же успехом использовать FAT или exFAT. **LXF**

MTP: совместное использование сети

Если вы уже пытались присоединить новый телефон Android или MP3-плеер к своему ПК через USB, вы, видимо, поняли, что их нельзя подмонтировать как устройства хранения, как мы делали в старые добрые времена. У этого есть основания — в частности, для телефона. Одной и той же файловой системы, подмонтированной сразу двумя системами, быть не может, поскольку одна из них в таком случае не сможет узнать, что делает другая. Вот почему Android раньше настаивал на размонтировании SD-карты в телефоне, перед тем, как разрешать вам монтировать ее на USB, что могло вызвать

проблемы у любого приложения, работающего с ней. Решение — использовать MTP (Media Transfer Protocol), являющийся формой совместной работы в сети. Файловая система остается под управлением устройства, но вы можете читать ее на своем компьютере, что во многом похоже на использование NFS или Samba share. Опять же мы обращаемся к FUSE в качестве решения для Linux, и есть, как минимум, три альтернативы: JMTPFS, Simple-mtpfs и JMTPFS. Я предпочитаю JMTPFS (блог Джейсона Феррары [Jason Ferrara] о его файловой системе MTP на базе Fuse см. здесь: <http://bit.ly/17z5dRA>).

Для ее использования надо всего лишь подключить ваш телефон и запустить команду

```
jmtarfs /mount/point
```

Она найдет ваш телефон и подмонтирует его по указанному пути. Если в телефоне более одной области хранения (например, встроенная и съемная карта), они отобразятся в виде отдельных директорий под точкой монтирования. Если те файловые системы, которые мы видим отовсюду, можно размонтировать стандартным способом, эта, как и другие файловые системы FUSE, размонтируется командой `fusermount -u /mount/point`

OMV: Рулим

Лучшие открытые NAS-решения — это для FreeBSD. Или нет? **Маянк Шарма** показывает NAS-решение для Debian, которое ничуть не хуже.



Наш эксперт

Маянк Шарма — ценитель данных, который покаялся не удалять ни файла, пока не увидит комету Галлея.



FreeNAS сделало себе имя как самое популярное и узнаваемое свободное решение для NAS (Network Attached Storage — сетевое хранилище данных). И хотя это решение пригодно для предприятия, с точки зрения средних домашних или офисных пользователей, которые хотят лишь хранить данные и эффективно управлять ими, последние релизы не в меру перенасыщены функциями.

Идеальное решение для таких пользователей — Open Media Vault (OMV). У программы нет экзотических требований к «железу», и ее можно установить на старый неиспользуемый компьютер всего с 1 ГБ ОЗУ. Если у вас несколько жестких дисков, OMV также может организовать их в RAID-массив.

Лучшее в OMV то, что программа основана на Debian. Ее интерфейс также более приспособлен для неспециалистов и поддерживает множество плагинов. Например, можно превратить NAS

в торрент-клиент для загрузки данных прямо в хранилище NAS, или использовать его для потокового вещания хранимой музыки.

Для начала воспользуйтесь инструкциями из пошагового руководства «Установка Open Media Vault», стр. 71, чтобы развернуть OMV. В оставшейся части руководства мы научимся добавлять диски в NAS и обращаться к ним с других компьютеров или устройств сети. Помните, что для первоначальной установки OMV понадобится физически подключить к компьютеру монитор и клавиатуру. После установки можно отключить их и управлять OMV с любого компьютера, находящегося в той же сети, что и сервер NAS.

Добавляем хранилище

Чтобы настроить сервер NAS, откройте web-браузер и введите IP-адрес компьютера, на котором запущена OMV. Для входа в web-интерфейс администратора OMV воспользуйтесь логином и паролем по умолчанию (имя пользователя: admin, пароль: openmediavault).

После входа в систему можно изменить пароль администратора по умолчанию, зайдя в System > General Settings [Система > Общие настройки] и перейдя на вкладку Web Administrator [Web-администратор]. В меню Settings [Настройки] также можно настроить другие аспекты сервера NAS — дата и время, брандмауэр, задания Cron, создать самоподписанные сертификаты SSL и др.

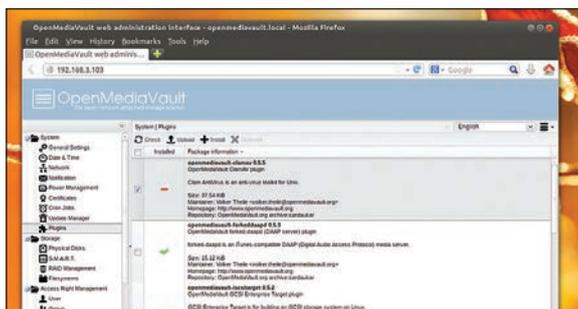
OMV может управлять несколькими физическими дисками или объединить их в устройство RAID, где для отказоустойчивости используются разные диски. Хотя по умолчанию используется RAID 5, OMV поддерживает все популярные форматы RAID. У каждого уровня RAID различное назначение и требования к дискам. Например, для RAID 1, при котором данные зеркалируются между дисками, необходимо не менее двух дисков, а для RAID 5 — не менее трех.

Зайдите в Storage > Physical Disks [Хранилище > Физические диски]; здесь перечислены все диски, подключенные к серверу. В списке также будет жесткий диск, на который вы установили OMV, но его использовать нельзя. На этой странице можно отформатировать диски, если вы будете использовать их по отдельности, а не в составе RAID. Выделите диск, на котором хотите хранить данные, и нажмите кнопку Wipe [Очистить]. Затем OMV предложит очистить диск быстро или надежно (т.е. более тщательно). После этого на нем можно создать файловую систему.

А если вы хотите объединить диски в RAID-массив, зайдите в Storage > RAID Management [Хранилище > Управление RAID] и нажмите кнопку Create [Создать]. В открывшемся диалоговом окне выберите устройства, которые хотите использовать в RAID, а также уровень RAID. В конце введите имя для устройства RAID и нажмите кнопку Save [Сохранить].

Перед созданием RAID нужно нажать зеленую кнопку в правом верхнем углу. На самом деле, зеленая кнопка будет загораться каждый раз при изменении конфигурации OMV, и ее нужно нажать, чтобы изменения вступили в силу. Для использования физических дисков или массива RAID нужно создать файловую систему. Перейдите в Storage > Filesystems [... > Файловые

» Для OMV есть плагин автоматической синхронизации разделяемого каталога с USB-устройством при его подключении.





ДАНЫМИ

системы] и нажмите кнопку Create. В открывшемся диалоговом окне выберите устройство, которое нужно отформатировать, из выпадающего списка, в котором также есть все устройства RAID, которые вы создали.

OMV поддерживает файловые системы ext3, ext4, XFS и JFS. По умолчанию он форматирует диски в ext4, но можно выбрать другую файловую систему из выпадающего меню. После выбора устройства и файловой системы для хранилища введите имя тома и нажмите кнопку Save.

Если вы пользуетесь несколькими дисками по отдельности, а не в RAID-массиве, не забудьте создать файловую систему на каждом из дисков. После создания файловой системы нажмите кнопку Mount [Смонтировать], чтобы ее активировать.

Делитесь файлами

Следующая задача — создать в NAS каталоги, в которых будут находиться данные. Можно создать один каталог или несколько — как вам необходимо. Зайдите в Access Rights Management > Shared Folders [Управление правами доступа > Разделяемые каталоги] и нажмите кнопку Add [Добавить]. В открывшемся диалоговом окне выберите из выпадающего списка том, в котором нужно создать каталог.

После выбора тома введите путь до каталога, который нужно сделать разделяемым, например, /documents. Поскольку диск был только что отформатирован и на нем ничего нет, указанный каталог создастся. Также укажите имя разделяемого каталога, например, Documents, и можете добавить комментарий для описания типа содержимого, которое будет находиться в каталоге.

Обратите особое внимание на параметр Permissions [Права доступа]. По умолчанию OMV разрешит читать и записывать данные в этот каталог только администратору и пользователям, которых вы добавите. Остальные пользователи смогут лишь читать его содержимое. Это довольно безопасный вариант для большинства систем, но вы можете выбрать более строгую или более свободную схему прав доступа из выпадающего списка.

Чтобы добавить пользователей в OMV, перейдите в Access Right Management > User [... > Пользователь]. Кнопка Add на этой странице представляет собой выпадающее меню, в котором можно добавить отдельных пользователей или импортировать список пользователей, добавив его в указанном формате.

При добавлении отдельного пользователя его также можно вписать в существующую группу. По умолчанию все пользователи добавляются в группу Users Group. Еще можно сделать так, что пользователь не сможет выполнять изменения в собственной учетной записи. Если вы хотите, чтобы у пользователей были свои домашние каталоги на сервере OMV, перейдите на вкладку Settings и поставьте галочку включения домашнего каталога для пользователя. Также потребуется указать расположение домашнего каталога, выбрав существующий разделяемый каталог на сервере NAS или создав новый.

Добавление пользователей дает вам еще одну возможность управлять доступом к разделяемым каталогам, и даже если вы при создании каталога выбрали права доступа по умолчанию, которые позволяют всем пользователям читать и записывать данные в каталог, можно запретить определенным пользователям читать или изменять его содержимое.

Потоковое вещание музыки

Если вы храните музыку в NAS, то сможете воспроизводить ее потоком на другие компьютеры и даже на свое устройство Android. Для этого сначала включите плагин forked-daapd, который добавляет возможность потокового воспроизведения музыки в OMV. В разделе System > Plugins [... > Плагины] вы увидите список плагинов, поддерживаемых OMV. Выберите плагин openmediavaultforkeddappd и нажмите кнопку Install [Установить]. После этого плагин будет загружен из онлайн-репозитория OMV.

Установив плагин, обновите web-интерфейс администрирования OMV. Тут вы

увидите, что в разделе Services [Сервисы] появился новый — iTunes/DAAP. Чтобы его настроить, укажите ему на разделяемый каталог NAS, который содержит музыкальные файлы. Теперь поставьте галочку Enable и нажмите кнопку Save, чтобы потоковый сервер запустился.

Для прослушивания музыки по сети используйте проигрыватель, который автоматически находит потоки DAAP и подключается к ним, такой как *Rhythmbox*, *Amarok*, *Banshee*, *Songbird*, *XBMC* и т. д. Для прослушивания потока на устройстве Android установите приложение DAAP Media Player с Google Play.

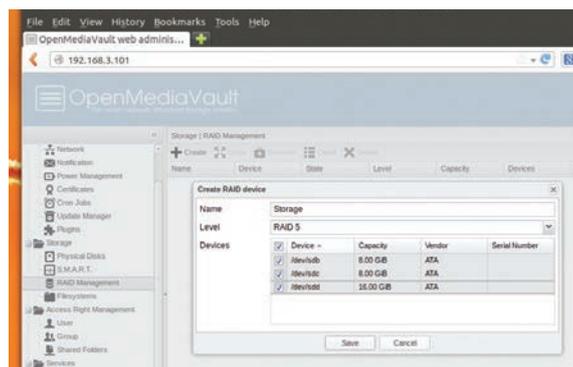
Для этого после добавления пользователя перейдите в раздел Shared Folders [Разделяемые каталоги], выберите каталог, права доступа к которому нужно изменить, и нажмите кнопку Privileges [Привилегии]. Откроется окно со списком пользователей, которых вы добавили, с галочками, управляющими их доступом к каталогу. Если вы что-то изменили, нажмите зеленую кнопку, чтобы изменения вступили в силу.

Настроив пользователей и разделяемые каталоги, переходите к организации общего доступа к хранилищу NAS в вашей сети. Единственное, что осталось сделать — включить сетевой сервис, который будет использоваться для доступа к разделяемым каталогам NAS пользователями.

OMV поддерживает различные протоколы и сервисы, включая NFS, SMB/CIFS, FTP, TFTP, SSH, *rsync* и другие. На нашем уроке мы будем использовать протокол SMB, широко известный под названием Samba. Он поддерживается всеми популярными операционными системами и работает с различными устройствами.

Для разделения каталогов с помощью Samba сначала нужно включить сервис в OMV. Перейдите в Servers > SMB/CIFS [Серверы > SMB/CIFS] и на вкладке Settings [Параметры] поставьте галочку Enable [Включить]. Другие параметры на этой вкладке можно не задавать. Когда закончите, нажмите кнопку Save для сохранения изменений.

»



» С OMV легко объединить несколько дисков в RAID-массив.

Старый NAS в новой бутылке

FreeNAS — самое популярное открытое NAS-решение, основанное на FreeBSD. Однако в более новых версиях FreeNAS рекомендуется не менее 8 ГБ ОЗУ, чтобы воспользоваться всеми возможностями файловой системы ZFS. Одна эта рекомендация делает FreeNAS неподходящим для большинства домашних пользователей. Кроме того, есть NAS4Free, которая тоже основана на FreeBSD, очень похожа на FreeNAS

с точки зрения функциональности и подходит для домашних и корпоративных пользователей.

NAS4Free — это не ответвление FreeNAS, а продолжение оригинала FreeNAS с 2011 года. На самом деле, текущий проект FreeNAS — ответвление исходного кода. Несмотря на то, что код был переписан, в нем продолжает использоваться прежнее название.

Под новым названием NAS4Free был первоначально выпущен в марте 2012 года. Вскоре после выпуска NAS4Free обновил основную систему с FreeBSD 7 на FreeBSD 9, что позволило дистрибутиву поддерживать множество новых устройств, включая новые процессоры Intel Atom. Перед установкой можно попробовать NAS4Free с помощью Live USB-брелка или карты CF.

После этого нужно добавить разделяемые каталоги как общие ресурсы Samba. Для этого перейдите на вкладку Shares [Общие ресурсы] и нажмите кнопку Add. В открывшемся окне выберите разделяемый каталог из выпадающего списка или кликните на зеленом плюсе, чтобы создать новый. Также нужно задать имя каталога. Помните, что под этим именем каталог будет отображаться в сети.

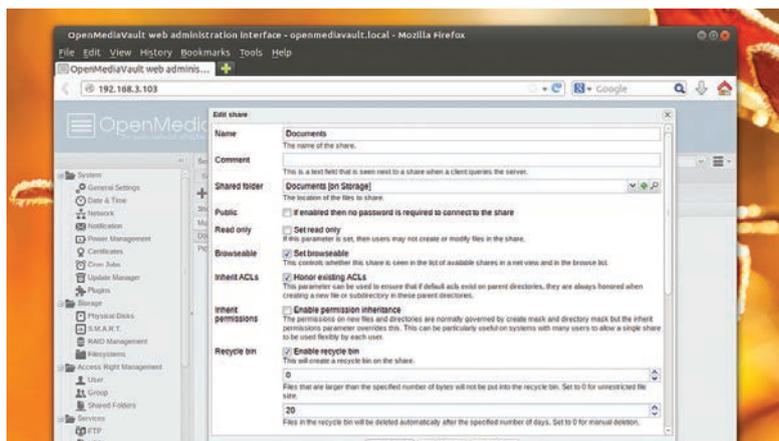
OMV при добавлении каталога Samba убедится, что для него соблюдены права доступа, заданные при создании разделяемого каталога в NAS. Если вы намереваетесь сделать каталог доступным всем, поставьте галочку Public [Публичный]. Если поставить галочку Set Read Only [Сделать доступным только для чтения], то ни один пользователь не сможет изменить содержимое каталога.

Один из параметров Samba может спасти вас в будущем — Recycle Bin [Корзина], так как по умолчанию она не включена, и если пользователь удалит файл, тот исчезнет из NAS навсегда. При добавлении каталога можно включить корзину; при этом удаленный файл будет перемещен в виртуальную корзину в разделяемом каталоге. Дополнительно можно задать срок, по истечении которого файлы удаляются из корзины навечно. Если у вас несколько разделяемых каталогов, их придется добавить как отдельные ресурсы Samba. Добавив все разделяемые каталоги в NAS, нажмите зеленую кнопку вверху, чтобы сохранить настройки и запустить сервис Samba.

Готово. Теперь у вас должен быть доступ к разделяемым каталогам NAS с любого устройства сети и из любимого файлового менеджера. Можно либо воспользоваться встроенной возможностью просмотра сетевых ресурсов файлового менеджера, либо ввести IP-адрес устройства NAS в строке адреса, например: <smb://192.168.3.103>.

Если вы не сделали ресурсы публичными, для доступа к ним у вас спросят логин и пароль. После проверки OMV смонтирует разделяемые каталоги. Теперь можно загружать файлы в разделяемые каталоги и удалять их, если у вас есть права доступа, как и в обычных каталогах.

➤ Не забудьте включить корзину при добавлении разделяемого каталога Samba.



Хотя для обмена данными с сервером NAS в основном будет использоваться сервис Samba, для взаимодействия с OMV можно включить и другие сервисы. Например, если вы хотите хранить в NAS резервные копии, включите сервис FTP в OMV. Все программы для резервного копирования — как простые вроде *Deja Dup*, так и сложные вроде *BackupPC* — позволяют сохранять резервные копии в удаленный каталог по FTP.

Включаем другие сервисы

Чтобы включить сервис FTP, перейдите в *Services > FTP*. Настройки FTP по умолчанию должны подойти большинству пользователей, и можно спокойно двигаться дальше, не изменяя их. Теперь перейдите на вкладку Shares и нажмите на кнопку Add, чтобы добавить разделяемый каталог для хранения резервных копий. Нужно просто выбрать каталог из списка разделяемых каталогов NAS; также можно добавить комментарий для описания содержимого каталога.

Следует только убедиться в том, что у пользователя есть права на чтение/запись в этот каталог. Чтобы проверить или изменить права доступа к разделяемому каталогу, зайдите в *Access Rights Management > Shared Folders*. Выделите каталог и нажмите кнопку Privileges, чтобы настроить права доступа для отдельных пользователей. После этого останется указать путь на устройстве NAS в приложении для резервного копирования.

Также стоит помнить, что хотя все параметры сервера NAS можно задавать в web-интерфейсе OMV, некоторые задачи, такие как обновление компонентов ядра, можно выполнять только с сервера. Чтобы облегчить удаленное управление с командной строки, лучше всего включить сервис SSH для удаленного подключения к серверу NAS.

Перейдите в *Services > SSH* и поставьте галочку Enable, затем нажмите кнопку Save. При желании также можно позволить пользователю root подключаться по SSH, поставив галочку Permit Root Login [Разрешить вход в систему root]. Также стоит убедиться, что версия OMV свежа. Для регулярных обновлений зайдите в *System > Update Manager [... > Менеджер обновления]* в web-интерфейсе. Выберите все перечисленные обновления и нажмите кнопку Install, которая загрузит их из онлайн-репозитория OMV.

Для выполнения крупных обновлений, которые затрагивают компоненты web-интерфейса и другие важные компоненты, вам нужно подключиться по SSH к NAS и выполнить команду `apt-get update` в командной строке. Если есть новый доступный релиз, можно переключиться на новую версию только командой `omv-release-upgrade`.

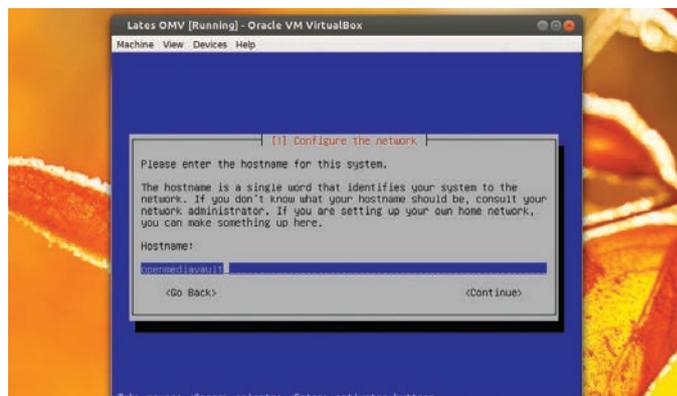
Open Media Vault — на удивление гибкое NAS-решение. Основная среда для получения совета и поддержки — форумы проекта, на которых также есть множество информации по выполнению установки в различных системах. Хотя программа не может похвастаться некоторыми возможностями других NAS-решений, таких как FreeNAS и NAS4Free, Open Media Vault идеально подходит для систем с умеренными требованиями, например, для дома и небольшого офиса. **LXF**

Установка Open Media Vault



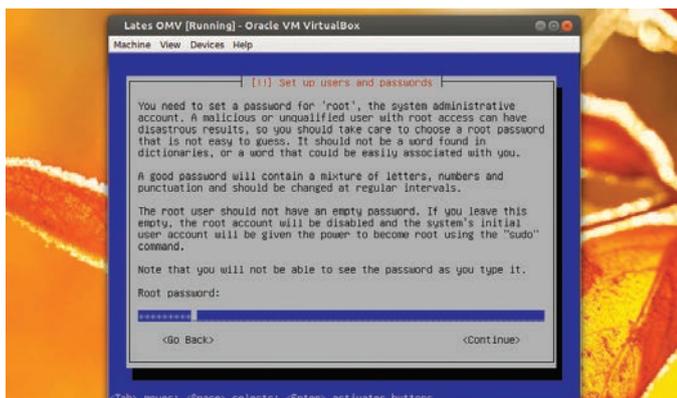
1 Прождем диск и загрузимся

Скопируйте 32- или 64-битный ISO-образ с **LXFDVD** или загрузите его с www.openmediavault.org. Запишите его на диск или скопируйте на флэшку с помощью *UNetbootin*. Загрузите с него компьютер, который скоро станет устройством NAS, и выберите Install в установочном меню.



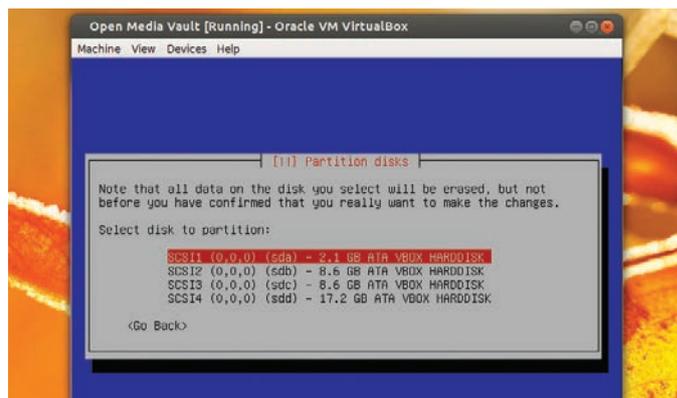
2 Настроим сеть

Во время установки вас попросят указать имя хоста и имя домена для устройства NAS. Имя хоста — одно слово, которое будет идентифицировать компьютер в сети. Если вы не знаете настроек своей сети, лучше оставить значения по умолчанию.



3 Пароль root

После настройки сети нужно указать пароль администратора NAS. Это пароль пользователя root, который отличается от административного пользователя, с которым вы заходите в административный интерфейс.



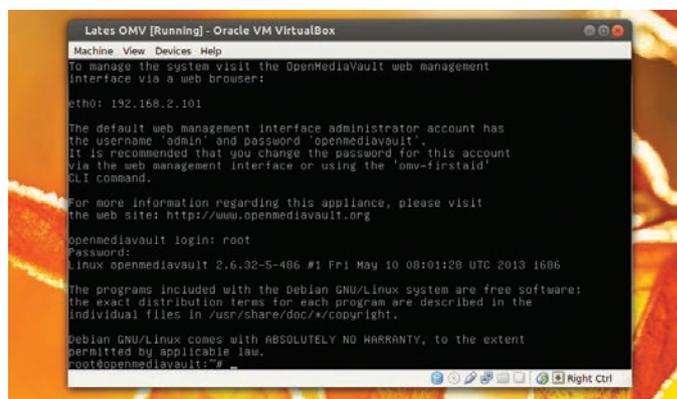
4 Выбираем установочный диск

OMV использует диск целиком, поэтому разбивать его на разделы не нужно. Просто укажите ей диск на 2 ГБ, этого достаточно. Также помните, что на установочном диске нельзя сохранять файлы, поэтому позаботьтесь установить OMV на самый маленький диск.



5 Выбираем репозиторий

Затем вас попросят выбрать одно из нескольких зеркал. OMV основана на Debian, и вам нужно регулярно загружать обновления из онлайн-репозитория, чтобы поддерживать систему обновленной.



6 Запускаем

Вот и все! После окончания установки выньте установочный диск и загрузитесь в OMV. Воспользуйтесь оболочкой и IP-адресом административного интерфейса для настройки своего устройства NAS.



Ядро Linux:

Не засиживайтесь на настройках по умолчанию — скомпилируйте свое ядро, с лучшей скоростью и функциональностью. **Майк Сондерс** — ваш проводник.



Наш эксперт

Майк Сондерс так любит копаться в операционных системах, что написал свою: <http://mikeos.berlios.de>.



Вопрос: какой программой в своем дистрибутиве Linux вы пользуетесь чаще всего? Вы, небось, назовете какой-нибудь Firefox или KDE, но не менее верным ответом будет «ядро». Ну да, вы не пользуетесь им напрямую — оно пытит где-то в фоне, позволяя всей системе работать — но без него вы не могли бы сделать ничего вообще. Ну, точнее, вы смогли бы полюбоваться своим сверкающим «железом» и уставить в пустой экран загрузчика, но веселого тут мало...

Так или иначе, хотя ядро — самая важная часть Linux, его обычно считают таким таинственным черным ящиком, где творится такое волшебство, что объяснить его могут только самые крутые умники. Даже если вы продвинутый пользователь Linux и следите за новостями ядра, вы наверняка никогда не пытались скомпилировать его сами. В конце концов, зачем эти сложности, если в дистрибутиве уже есть готовое ядро? А вот зачем...

» Многие стандартные ядра в дистрибутивах оптимизированы для работы на большом диапазоне устройств. Скомпилировав свое ядро, вы сможете пользоваться оптимизациями для вашего конкретного процессора, получив прирост скорости.

» Некоторые функции в исходном коде ядра помечены как экспериментальные и по умолчанию не включаются в ядра в дистрибутивах. Скомпилировав свое ядро, можно обзавестись этими функциями.

» На просторах Интернета гуляет множество полезных заплаток ядра, которые можно применить к исходному коду и создать дополнительные возможности.

» Да просто из любопытства. Компиляция и установка нового ядра — отличный способ проникнуть в недра Linux.

Итак, на нашем уроке мы покажем, шаг за шагом, как получить, настроить, скомпилировать и установить новое ядро. Мы также рассмотрим, как применять заплатки из Интернета. Но, пожалуйста, обратите внимание на БОЛЬШОЕ ЖИРНОЕ ПРЕДУПРЕЖДЕНИЕ: установка нового ядра — нечто вроде операции на мозге для компьютера. Это увлекательно, но... бывают и осложнения. Мы не несем ответственности за то, что вы угробили свою систему! Поэтому настоятельно рекомендуем делать это в системе, с которой не жалко поэкспериментировать, или в виртуальной машине.

Подготовка

Первое, что нужно сделать — загрузить исходный код ядра. В разных дистрибутивах используются разные версии ядра, и в большинстве из них наложены дополнительные заплатки, но в данном случае мы воспользуемся чистым подходом — возьмем исходный код, который одобрил лично Линус Торвальдс.

Домашняя страничка разработчиков ядра в Интернете — www.kernel.org, и оттуда можно загрузить последний официальный релиз. На нашем уроке мы воспользуемся версией 3.10.9 из файла **linux-3.10.9.tar.xz**; когда вы будете читать эту статью, может появиться версия поновее. Тогда загрузите ее и следуйте уроку, просто изменив номер версии, где это необходимо.

Далее, вместо распаковки исходного кода в домашний (или временный) каталог гораздо лучше распаковать его в **/usr/src**. Сейчас это не критически важно, но позже может понадобиться: некоторым программам нужны файлы заголовков (**.h**) для текущего ядра, и они часто ищут исходный код в **/usr/src**. Один из примеров — *VirtualBox*: у этой программы собственный модуль ядра, и для сборки этого модуля во время установки ей нужны заголовки исходных файлов ядра.

Поэтому распакуйте исходный код следующей командой (но учтите, что все команды на нашем уроке должны выполняться с правами root):

```
tar xfv linux-3.10.9.tar.xz -C /usr/src/
```

Распаковка ядра потребует времени, так как последние версии исходников ядра весят почти 600 МБ. Зайдите в созданный каталог командой **cd /usr/src/linux-3.10.9** и осмотритесь в нем командой **ls**.

Мы не будем объяснять назначение всех каталогов, которые вы здесь увидите, так как это тема для совсем другой статьи, но, возможно, вам будет любопытно заглянуть в некоторые из них. Например, в каталоге **mm** вы найдете код менеджера памяти, тогда как **arch/x86/kernel/head_32.S** — это сборка для выполнения кода 32-битных ПК. Здесь ядро делает свою, так сказать, первоначальную работу.

Перейдем к самому интересному этапу настройки ядра: его подстройке под вашу систему. В каталоге **/usr/src/linux-3.10.9** введите:

```
make xconfig
```

Создаем свое

Если у вас установлены файлы для разработки Qt 4 (например, пакет *libqt4-dev* в Debian/Ubuntu), эта команда скомпилирует и запустит графическую утилиту настройки. Для GTK-альтернативы попробуйте следующую команду:

```
make gconfig
```

и если ни одна из графических версий не запустится, есть вполне приличная текстовая утилита (для нее необходима *Ncurses*):

```
make menuconfig
```

Несмотря на различие интерфейсов, все эти утилиты настройки имеют одни и те же параметры. И параметров много — буквально тысячи. Если вы читаете эту статью холодным зимним вечером и у вас есть немного свободного времени, заварите себе чайку и пройдитесь по некоторым категориям.

Правда, большая часть этой информации сугубо техническая и относится к очень узким настройкам устройств или системы, но просто пролистав настройки, вы увидите, каким гибким и богатым возможностями является ядро Linux. И вы поймете, почему оно используется везде — от мобильных телефонов до суперкомпьютеров.

Приукрашим ядро

На нашем уроке мы сосредоточимся на утилите *Xconfig*, поскольку у нее самый простой и удобный интерфейс. В левой части вы видите дерево параметров и категорий — при щелчке по категории на верхней правой панели появятся ее параметры. Если теперь щелкнуть на одном из них, на нижней правой панели возникнет соответствующая подсказка. Большинство параметров ядра хорошо задокументированы, так что щелкайте по ним и наслаждайтесь исследованием.

Включенные параметры отмечены знакомыми галочками; также есть много квадратиков с кружками внутри. Это означает, что выбранный параметр будет собран в виде модуля, то есть в само ядро не войдет, но загрузится по требованию. Если вкомпилировать все возможности в ядро, то файл ядра разбухнет до огромного размера, возможно, даже непосильного для загрузчика. Следовательно, лучше включать в ядро только критически

важные функции и драйверы, а все остальное (например, функции, которые можно включить после загрузки системы) оставить в виде модулей.

Поэтому ставьте галочки для изменения состояния между включенным (пустая клетка), включенным в ядро (галочка) или скомпилированным как отдельный модуль (кружок). Обратите внимание, что некоторые возможности нельзя скомпилировать как отдельные модули, и они либо включены, либо нет. Возможно, сейчас вы спрашиваете: а с чего бы это? Кто принял все эти решения? Ну, посмотрев в окно терминала, в котором вы запускали **make xconfig**, вы увидите следующую строку:

```
# using defaults found in /boot/config-3.8.0-21-generic
```

К вашему текущему ядру привязан файл настройки в каталоге */boot*, и утилиты *xconfig/gconfig/menuconfig* находят его и берут за основу для новой конфигурации. Это прекрасно, так как новое ядро будет иметь похожий набор возможностей по сравнению с текущим — что снижает шансы поиметь зрелищные катастрофы при загрузке.

Когда вы нажмете кнопку *Save [Сохранить]* в утилите настройки, та сохранит параметры в *.config*, и при любом последующем использовании *xconfig/gconfig* и т.д. будет использоваться этот файл *.config*.

Включение добавочных функций

В начале этой статьи мы говорили об изменении ядра ради улучшения производительности и использования экспериментальных возможностей. Для первого загляните в тип *Processor [Процессор]* и категорию *“features [функции]”*. Ваше текущее ядро, вероятно, было настроено под *Pentium Pro* или другой старый процессор — по сути, тут нет ничего худого, поскольку благодаря этому ядро будет работать на большем диапазоне процессоров; но вы, вероятно, захотите выбрать нечто поновее. Например, если у вас процессор *Core i3/i5/i7*, выберите опцию *“Core 2/newer Xeon”*. Огромного скачка скорости ждать не стоит, но, по крайней мере, ядро будет скомпилировано с оптимизациями для последних процессоров Intel.

»

Один день из жизни ядра

Если вы новичок в мире Linux или особо не вникали в технические детали своей операционной системы, вы, вероятно, знаете только, что ядро — ключевая часть системы и делает всю важную работу. Но что именно оно делает?

Чтобы прояснить это, рассмотрим ключевые задачи ядра немного подробнее:

» **Выполнение программ** Ядро по сути является «боссом» для всех работающих программ. Вы не можете отдать какой-то одной программе все ресурсы процессора, и если эта программа зависает, вам не надо, чтобы зависли и все остальные. Поэтому ядро выдает программам отдельные доли

процессорного времени, гарантируя, что все они будут работать вместе, и ни одна из них не получит полный контроль над компьютером. Ядро также может завершать запущенные программы и освобождать их ресурсы.

» **Доступ к устройствам** Лишь очень немногие пользовательские программы работают с устройствами напрямую. Вы вряд ли захотите, чтобы две программы пытались одновременно использовать, например, один и тот же USB-порт, вызывая всевозможные конфликты. Поэтому ядро работает с устройствами, предоставляя драйвера для доступа к конкретным устройствам, а также уровни

абстракции, благодаря которым программы более высокого уровня не обязаны знать подробностей работы каждого устройства.

» **Управление памятью** Только представьте, что каждая программа вольна распоряжаться в ОЗУ. Программы не будут знать, какой частью памяти кто пользуется, начнут наткаться друг на друга, и в итоге открытый вами документ внезапно запоет звуковыми данными. Ядро выделяет участки памяти программам и гарантирует их сосуществование отдельно друг от друга, и если какая-то программа выйдет из-под контроля, ей будет не повредить область памяти другой программы.

» **Не хотите пропустить номер?** Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Перейдем к экспериментальным возможностям ядра: при навигации в категориях вы увидите, что некоторые параметры помечены словом “Experimental [Экспериментальный]” или “Dangerous [Опасный]”. Достаточно сказать, что это не те параметры, на которые можно положиться, так как в них будут возникать ошибки, и им нужно время на созревание. Но если вам отчаянно хочется попробовать новейшую возможность, о которой вы где-то прочли, ищите здесь.

Время сборки!

После необходимой настройки параметров ядра сохраните их и выйдите из утилиты настройки. Теоретически теперь ядро можно собрать одной командой **make**, но для многоядерных процессоров такой способ неэффективен: в этом случае лучше указать параметр **-j** с количеством ядер процессора. Этот параметр велит **make** разбить компиляцию на несколько параллельных задач, что значительно уменьшит общее время сборки ядра. Так, если у вас двухядерный процессор, воспользуйтесь командой

```
make -j 2
```

Время сборки ядра зависит от количества функций, которые вы включили в ядро, и от аппаратных спецификаций компьютера. Например, если вы собираете довольно урезанное ядро на последнем процессоре Core i7, оно будет готово минут за 15. Если у вас более старый компьютер и вы собираете ядро, в котором имеется все, включая посудомойку, сборка способна занять несколько часов.

В любом случае, когда процесс будет закончен, настанет время установить ядро и модули туда, где они должны находиться:

```
make modules_install
make install
```

Очень важно вводить команды именно в этом порядке. Первая команда помещает модули ядра в каталог `/lib/modules/<версия_ядра>`, в нашем случае, `/lib/modules/3.10.9/`. Затем вторая команда копирует ядро и его вспомогательные файлы в каталог `/boot`. Вот эти файлы:

» **vmlinuz-3.10.9** Сжатый образ ядра. Он вызывается загрузчиком *Grub* и выполняется.

» **system.map-3.10.9** Таблица символьных имен (например, имен функций) и их адресов в памяти. Она удобна для отладки в случае падений ядра.

» **initrd-3.10.9** «Первоначальный RAMdisk [initial RAMdisk]» — небольшая корневая файловая система с набором драйверов и утилит, необходимых для загрузки системы (и монтирования настоящей файловой системы из другого места).

» **config-3.10.9** Копия файла `.config`, созданная при запуске **make xconfig** или одного из ее вариантов.

Процесс **make install** также услужливо обновляет загрузчик *Grub*. Заглянув в файл `/boot/grub/grub.cfg`, вы увидите в нем новые записи для версии ядра, которую только что скомпилировали и установили.

Теперь вы готовы к самому волнующему моменту: перезагрузке в свое новое сияющее индивидуальное ядро! Просто выберите его в меню *Grub*, скрестите пальцы и смотрите, что происходит. Если все прошло удачно, ваша система загрузится корректно и вы сможете исследовать новые возможности, которые добавили; но если нет, не беспокойтесь. Просто загрузитесь снова и выберите старое ядро (оно скорее всего будет находиться в подменю “Advanced options [Дополнительные варианты]”). Linux прекрасно поддерживает несколько версий ядер на одном компьютере, поэтому ваша система вряд ли станет незагружаемой.

Если нужно внести в ядро дальнейшие изменения, просто еще раз запустите **make xconfig** и следуйте описанной выше процедуре. Также можно удалить скомпилированные файлы командой **make clean**, но при этом `.config` и некоторые другие файлы остаются на месте. Для сброса исходников в их исходный, нетронутый вид выполните команду:

```
make mrproper
```

После выполнения этой команды все временные файлы и настройки будут удалены, как будто вы только что распаковали архив со свежими исходниками.

Золотая мяня

Прекрасный способ украсить свое ядро дополнительными функциями — воспользоваться одной из множества заплат, доступных в Интернете (см. врезку «Заплаты, на которые стоит взглянуть» на стр. 75). Здесь мы сосредоточимся на том, как применить заплатку. В данном случае мы возьмем заплатку реального времени в файле **patch-3.10.9-rt5.patch.gz**. Мы загрузили ее в каталог `/usr/src/linux-3.10.9`, и, как следует из имени файла, это один файл `.patch`, сжатый посредством *gzip*.

Заглянув внутрь файла (например, командой **zless patch-3.10.9-rt5.patch.gz**), вы увидите несколько строк, начинающихся с плюсов (+) и минусов (-). Если вкратце, строки с плюсами добавляются к исходному коду ядра в ходе применения заплатки; с минусами — удаляются оттуда. Между каждыми фрагментами, помеченными словом “diff”, вы видите строки “+++” и “---”, они показывают, какие файлы будут изменены.

Можно тут же и залатать исходный код, но лучше сначала сделать тестовый прогон и убедиться, что все пройдет нормально. К счастью, у команды латания есть параметр, который это самое и делает:

```
zcat patch-3.10.9-rt5.patch.gz | patch -p1 --dry-run
```

Здесь мы распаковываем сжатую заплатку в стандартный поток вывода (т.е. в терминал) и затем направляем ее содержимое утилите *patch*. Параметр **-p1** показывает, что мы хотим применить

заплатку внутри текущего каталога, а **--dry-run** запускает команду в режиме, в котором все изменения просто выводятся на экран, а не выполняются на самом деле. На экране вы должны увидеть множество строк вроде:

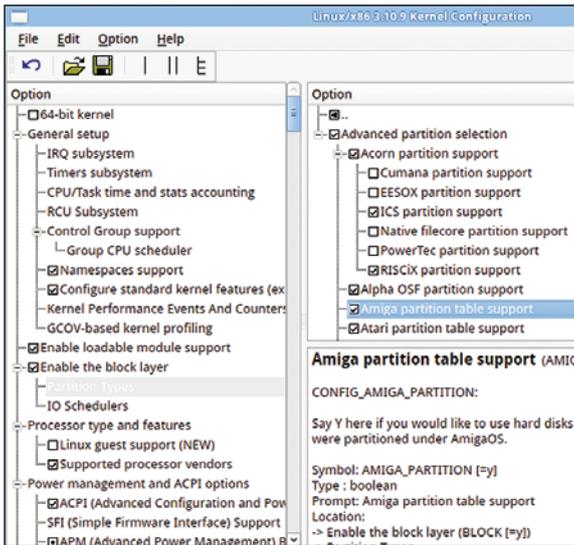
```
patching file arch/sh/mm/fault.c
```

Если все пройдет хорошо и не вылезут гремлины, повторите команду без параметра **--dry-run**. Если вы используете заплатку,



» Большая желтая кнопка на www.kernel.org всегда указывает на последний стабильный релиз исходников ядра.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



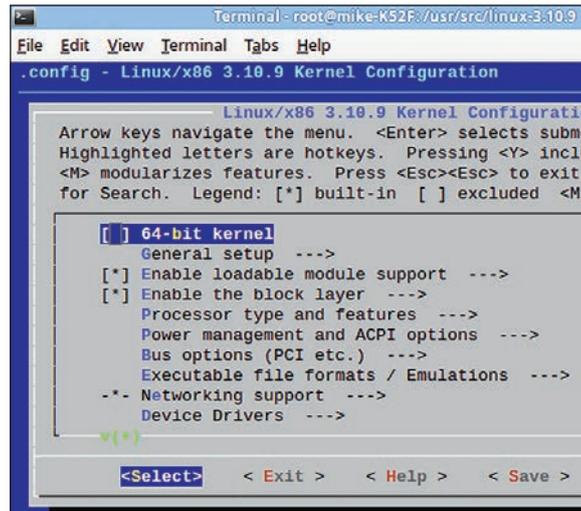
» **Xconfig** предоставляет графический интерфейс для настройки ядра на базе *Qt*...

название которой оканчивается на **.bz2**, используйте в начале команды **bzcat**, а для файлов XZ используйте **zxcacat**.

Подлатав ядро, вернитесь обратно в утилиту настройки ядра, чтобы включить новые возможности (если это необходимо), и повторите сборку как описано выше. После этого приходите на следующую встречу LUG с широкой улыбкой на лице, рассказывая всем, как вы используете потрясающее ядро ручной настройки со свежайшими заплатками из Интернета. Может, пивком кто угостит...

Создайте заплатку сами

Если вам страстно хочется взломать ядро самим, первое, что нужно сделать — добавить в него нечто полезное. Эта тема целиком выходит за рамки нашего урока, но чтобы поделиться своей работой с миром, вам также нужно создать файл **.patch**,



» ...но если вы подключились по SSH к серверу без *X Window*, воспользуйтесь вместо этого программой **menuconfig** на основе *Ncurses*.

который показывает разницу между исходным ядром и вашим форсированным.

Для этого вам нужны два каталога: один с исходным нетронутым кодом ядра и второй с вашими изменениями. Так, в **/usr/src** у вас могут быть каталоги **linux-3.9.10** и **linux-3.9.10-me**; во втором находится измененный код. Чтобы сгенерировать файл заплатки, скомануйте

```
diff -uprN linux-3.10.9/ linux-3.10.9-me/ > myfile.patch
```

Теперь можно упаковать заплатку и поделиться ею с остальными, чтобы они могли применить ее к своему ядру, пользуясь инструкциями, описанными выше.

Если вы сделали что-то по-настоящему клевое и хотите отправить это разработчику ядра, обязательно прочтите файл **Documentation/SubmittingPatches** в исходном коде ядра — в нем есть много полезных советов. **LXF**

Запатки, на которые стоит взглянуть

Окончательное решение о том, что войдет в ядро Linux, принимает Линус Торвалдс, и за последние несколько десятков лет он доказал, что является хорошим менеджером проектов (разве что иногда чересчур многословным). Следовательно, никаких крупных разветвлений ядра из-за ухода разработчиков не было, но отдельные ветви кода ядра существуют. Они включают экспериментальные функции или развивают ядро в другом направлении, по сравнению с предложенным Торвалдсом и другими, и самые известные из них включают:

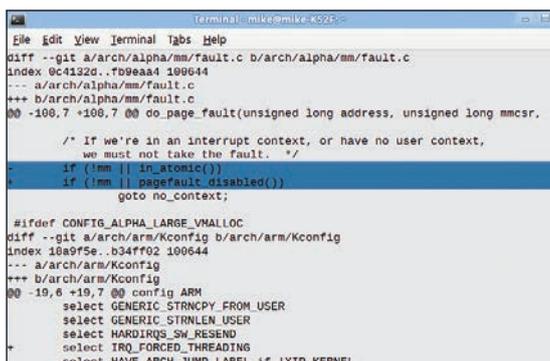
» **pf-kernel** (<http://pf.natalenko.name>) Несколько «замечательных» возможностей, не включенных в основную ветку» (официальное дерево исходников ядра). Оно включает альтернативный планировщик ввода/вывода, *TuxOnIce* для улучшения гибернации и заплатку **-ck** для улучшения отклика.

» **TinyLinux** (www.tinylinux.org/project/tinylinux) Этот набор заплат разработывался для встраиваемых систем с ограниченными ресурсами, и в нем приложены усилия по снижению потребления жесткого диска и оперативной памяти ядром.

» **RT Patch** (<https://rt.wiki.kernel.org>) Стандартное ядро Linux не блестяще работает в режиме реального времени, так как в большинстве серверных и настольных систем не имеет особого значения, если для завершения какой-то задачи понадобится еще 0,01 с, когда система находится под нагрузкой. Но если вы профессионально работаете со звуком или используете Linux

для управления механизмами, то захотите гарантировать, чтобы ядро выполняло определенные операции в точные сроки (чтобы сохранить синхронизацию), и RT Patch предоставляет такую возможность.

Учтите, что для синхронизации многих наборов заплат с последней версией исходников ядра требуется время, и для той версии ядра, которую вы собираете, заплатки может не найтись.



» Вот так выглядит RT Patch — выделение показывает, как добавляются и удаляются строки.



Ваш первенец в

Джоно Бэкон знакомит нас с мощным SDK Ubuntu для построения мобильных, планшетных и настольных приложений.



Наш эксперт

Джоно Бэкон управляет сообществом Ubuntu, он автор книги «Искусство сообщества» и основатель ежегодного Саммита лидеров сообщества.

Ядро стратегии конвергенции Canonical — совершенно новый Ubuntu Software Development Kit (SDK — набор средств для разработки ПО), который содержит полноценное окружение разработки с интегрированной средой разработки и поддерживает запуск программ на рабочем столе и на телефонах и планшетах Ubuntu. С SDK можно писать программы с использованием различных фреймворков: QML (на основе Qt), HTML5, Scores (способы поиска и представления контента в Ubuntu Dash) и OpenGL.

На нашем уроке мы напишем приложение на QML, которым пользуется большинство разработчиков новых приложений Ubuntu. Сначала мы рассмотрим, что такое QML и как он работает, а затем я покажу, как написать приложение под названием *Sleepy Time*, которое я написал и загрузил в Ubuntu Software Centre. Это приложение проигрывает успокаивающие звуки, которые помогают младенцам, детям и взрослым заснуть.

Создание проекта

Сперва загрузим SDK Ubuntu и создадим новый проект QML. Для этого выберите File > New File Or Project [Файл > Новый файл или проект]. Теперь убедитесь, что на левой панели выбрана Ubuntu, и щелкните на Simple Touch UI, а затем на Create [Создать]. В следующем окне введите имя проекта — 'sleepy', и выберите место, в котором будет создан проект. В выпадающем списке Add To Version Control [Добавить в систему управления версиями] выберите Vazaar, чтобы мы могли работать с проектом в системе управления версиями, и затем нажмите кнопку Finish [Готово].

Создаст несколько файлов, которые показаны на левой панели. По двойному щелчку на файле он загружается в текстовый редактор. Дважды щелкните на `sleepy.qml`, если он еще не загружен, и вы увидите в редакторе следующий код:

```
import QtQuick 2.0
import Ubuntu.Components 0.1
import "components"
MainView {
    objectName: "mainView"
    applicationName: "Ixfstest"
    width: units.gu(100)
    height: units.gu(75)
    Page {
        title: i18n.tr("Simple")
        Column {
            spacing: units.gu(1)
            anchors {
                margins: units.gu(2)
                fill: parent
            }
            HelloComponent {
                id: label
                objectName: "label"
                text: i18n.tr("Hello..")
            }
            Button {
```

```
objectName: "button"
width: parent.width
text: i18n.tr("Tap me!")
onClicked: {
    label.text = i18n.tr("..world!")
}}}}
```

Теперь можно запустить проект, нажав на зеленую стрелку на левой панели ближе к нижней части окна. При запуске приложения появляется кнопка Tap Me [Нажми меня], которая при нажатии меняет текст в поле выше.

В начале этого фрагмента кода мы импортируем ряд библиотек с функциями, как и в других языках программирования. Важнейшая библиотека здесь — *Ubuntu.Components*. Это набор элементов пользовательского интерфейса, используемых для построения приложений. Чтобы увидеть весь набор доступных компонентов, выберите Tools > Ubuntu Touch > Ubuntu Touch Showcase Gallery [Инструменты > Ubuntu Touch > Витрина компонентов Ubuntu Touch]. Тогда загрузится новый проект, и вы сможете увидеть компоненты, нажав на зеленую стрелку, а также просмотреть код. Закончив высматривать компоненты Витрины, щелкните правой кнопкой мыши на верхней записи "sleepy" в боковой колонке (с синей папкой и стрелкой влево) и выберите Set sleepy as Active Project [Установить sleepy как активный проект], чтобы продолжить работу над этим проектом (так мы гарантируем, что при запуске приложения запустится `sleepy.qml`, а не витрина компонентов).

В QML у нас есть набор вложенных компонентов пользовательского интерфейса; каждое вложение выполняется с помощью фигурных скобок (`{}` и `}`). Корневой контейнер, используемый во всех приложениях — `MainView` {}; в него вложен контейнер `Page`{} — это область, которую можно заполнить содержимым своего приложения. Внутри `Page`{} у нас есть элемент `Column`{}, который просто располагает вложенные элементы вертикально. Внутри `Column` сидит нечто под названием `HelloComponent`{}. Это компонент, определенный в другом месте (в подкаталоге 'components'). Имя файла компонента должно соответствовать имени компонента (например, для `HelloComponent`{} это `components/hellocomponent.qml`). Такой метод вложения компонентов представляет удобный способ разделения и повторного использования кода наших приложений, что мы обсудим позже.

Под компонентом `HelloComponent`{} у нас есть `Button`{} — это еще один компонент Ubuntu, который просто отображает кнопку, на которую мы нажимаем для изменения текста `HelloComponent`{}.

Внутри каждого компонента есть параметры, которые можно менять. Например, в `Button`{} мы задаем имя объекта, к которому затем можно обратиться (например, `objectName`), и текст, отображаемый на кнопке (`text`). Заметьте также, что при задании текста мы обернули его в `i18n.tr()`: тогда строка будет отмечена как переводимая, и мы сможем перевести ее с `gettext`.

Ключевая возможность набора инструментов пользовательского интерфейса Ubuntu [Ubuntu User Interface Toolkit] — возможность масштабирования на все форм-факторы в мире, заданные пользователями с различными устройствами. С помощью этого подхода был определен новый тип единицы — единица

Скорая помощь



SDK Ubuntu невероятно прост в установке. Полный SDK доступен в Ubuntu Software Centre. Просто наберите в поиске "Ubuntu SDK" и нажмите «Установить». После этого можно запустить SDK, найдя его в Ubuntu Dash и щелкнув на его иконке.

SDK Ubuntu

Шкала разрешений

Устройство	Преобразование
Большинство ноутбуков	1 gu = 8 пикселей
Ноутбуки с дисплеями Retina	1 gu = 16 пикселей
Смартфоны	1 gu = 18 пикселей

➤ Подход Ubuntu к изменяемому дизайну.

сетки (grid unit или, сокращенно, gu). Единицы сетки преобразуются в значения пикселей в зависимости от типа экрана и устройства, где запущено приложение (см. врезку вверху).

В коде выше мы задали расстояние между дочерними элементами **Column** в 1 gu и задали границы полей как часть якорей (границы прикрепляются к родительскому компоненту **Page**). Мы также задали размер самого приложения параметрами **width** [ширина] и **height** [высота] в **MainView**{}.

Компоненты и сигнал

Перед тем как двигаться дальше, давайте разберем **HelloComponent**{}. Код в файле **components/hellocomponent.qml** выглядит так:

```
import QtQuick 2.0
import Ubuntu.Components 0.1

UbuntuShape {
    width: 200
    height: width
    property alias text : myText.text
    Label {
        id: myText
        anchors.centerIn: parent
    }
}
```

Здесь мы используем компонент **UbuntuShape**{} (квадратик с приятно скругленными углами), встраиваем в него компонент **Label**{}, id которого — **myText**, и выравниваем текст по центру. Возможно, вы заметили, что сам текст мы не задаем. Для этого нам нужны свойства.

У компонента **Label**{} есть свойство **text**, с помощью которого можно задать текст метки, но мы хотим задать текст вне **HelloComponent**{} и в другом исходном файле. Поэтому нам нужна такая строка:

```
property alias text : myText.text
```

Она создает алиас для свойства **myText.text**, который доступен вне компонента по алиасу **text**. Чтобы посмотреть это в действии, вернитесь в **sleepy.qml**, и увидите следующее:

```
onClicked: {
    label.text = i18n.tr("!.world!")
}
```

Этот блок появляется внутри компонента **Button**, и вот так в QML компоненты взаимодействуют друг с другом.

По сути, у каждого компонента есть набор различных сигналов, при возникновении которых можно быть выполнен некоторый код. В данном случае, у компонента **Button** есть сигнал «нажато [clicked]», и когда мы используем блок **onClicked** (при доделывании «on» к названию сигнала мы получаем имя блока,



который отвечает на сигнал), то код в этом блоке выполняется при нажатии кнопки.

Так уж вышло, что этот код ссылается на объект метки (наш компонент **HelloComponent**{}), и в нем мы устанавливаем свойство **label.text**, наш заданный ранее алиас для текста, который должен появиться при нажатии кнопки.

Четко разобравшись в нашей программе и ключевых основах QML, сделаем наше приложение более полезным. Мы создадим простой звуковой проигрыватель, который будет проигрывать несколько атмосферных звуков, которые помогут людям, особенно малышам, заснуть.

Для начала удалите из **sleepy.qml** весь блок **Column** и задайте название [title] компонента **Page** “Sleepy”. Теперь, запустив приложение, вы должны увидеть пустое окно с заголовком “Sleepy”. Мы очистили окно, чтобы разместить на нашей сетке звуки.

Звуки будут представлены в виде нескольких квадратов в стиле Ubuntu, на каждом из которых будет иконка и какой-то текст, и при нажатии на квадрат начнет проигрываться мелодия, а цвет квадрата изменится, чтобы показать, что звук проигрывается. В нашем приложении можно будет одновременно проигрывать несколько звуков при нажатии нескольких кнопок.

Для получения такого результата нам нужно создать собственный компонент для кнопки, проигрывающей звук, точно так же, как мы делали с **HelloComponent**{}. Для этого выберите **File > New File Or Project** [Файл > Новый файл или проект], и на левой панели выберите **Qt** и затем **QML File (Qt Quick 2)**. Назовите

➤ По щелчку на ссылке API слева отображается информация об API компонентов Ubuntu.

Публикация вашего приложения

Когда вы напишете свое первое приложение и захотите поделиться им со всем миром, передать его в руки пользователей Ubuntu очень просто. Недавно Canonical выпустила бета-версию нового процесса публикации приложений Ubuntu Touch.

Для публикации приложения достаточно сгенерировать пакет (в SDK Ubuntu

делается несколькими щелчками) и загрузить его на <http://developer.ubuntu.com>, куда можно добавить информацию о приложении, несколько скриншотов, иконку и дополнительные подробности.

Подробная информация о публикации приложений Ubuntu приведена на <http://developer.ubuntu.com/publish>.

➤ **Финальная версия нашего приложения *Sleepy* с иконками и градиентом в фоне.**



файл **SoundButton.qml** (регистр учитывается) и добавьте его в проект. Теперь добавьте в этот файл следующий исходный код:

```
import QtQuick 2.0
import QtMultimedia 5.0
import Ubuntu.Components 0.1
UbuntuShape {
    id: box;
    antialiasing: true;
    radius: "medium"
    property alias color: box.color;
    property alias description: label.text;
    property alias imageSource: image.source;
    property alias soundSource: sound.source;
    property var state: false;
    Image {
        id: image;
        anchors.horizontalCenter: parent.horizontalCenter;
        anchors.verticalCenter: parent.verticalCenter;
        width: parent.width * 0.8;
        height: parent.height * 0.8;
        fillMode: Image.PreserveAspectRatio
    }
    Label {
        id: label;
        anchors.horizontalCenter: image.horizontalCenter
        anchors.top: image.bottom
        text: "Hello, world!"
        fontSize: "medium"
    }
    Audio {
        id: sound;
        onStopped: {
            box.color = "#32222C"
            if (box.state == true) {
                box.color = UbuntuColors.warmGrey sound.play()
            }
        }
    }
}
MouseArea {
```

```
anchors.fill: parent;
onPressed: {
    if (box.state == false) {
        box.state = true;
        sound.play();
        box.color = UbuntuColors.warmGrey
    }
    else if (box.state == true) {
        box.state = false;
        sound.stop()
        box.color = "#32222C"
    }
}
```

Хотя кажется, что кода очень много, большая его часть должна быть знакомой на основе того, что мы уже освоили. Пройдемся по нему шаг за шагом.

Сначала мы указываем импортируемые библиотеки, и здесь есть одна новая: *Qt.Multimedia*. Она предоставляет набор функций для работы с видео и звуком. Мы включили ее, чтобы проигрывать звуковые файлы при нажатии на кнопки.

Затем мы создаем нашу кнопку **SoundButton**; она состоит из **UbuntuShape**{}, внутри которой — **Image**{ } и **Label**{ } под ним. Внутри компонента **Image**{ } есть несколько свойств, которые привязывают изображение к центру **UbuntuShape**{}, а затем с помощью свойств **width** и **height** добавляется небольшой ободок вокруг изображения. В **Label**{ } мы задаем некий текст, который позже будет заменен другим, а также размер шрифта и выравнивание.

Воспроизведение звука

Теперь посмотрим, как в этом компоненте работает звук. Когда пользователь нажимает на кнопку, нам нужно не только проиграть звук, но и остановить воспроизведение при повторном нажатии. Поэтому необходимо следить за тем, проигрывается ли звук в данный момент или нет. Кроме того, такие приложения обычно проигрывают звук всю ночь, пока вы спите, и каждую мелодию нужно воспроизводить в цикле, чтобы она не прекращалась после первого воспроизведения.

Начнем с добавления компонента **Audio**{ } в **UbuntuShape**{}, который только лишь указывает, что в этом компоненте должен проигрываться звук. Прежде чем перейти к воспроизведению, взглянем на строку в свойствах верхнего уровня **UbuntuShape**{}, которую вы, возможно, уже заметили:

```
property var state: false;
```

Это переменная, с помощью которой мы будем отслеживать, проигрывается ли звук, в каждом экземпляре звуковой кнопки. По умолчанию мы установили ее в "false", так как при создании компонента звук не проигрывается.

Теперь посмотрим, как работает воспроизведение. Сперва мы создаем компонент **Audio**{}, в котором нет ничего особенного. Присутствие этого компонента в **UbuntuShape**{ } означает, что наш компонент сможет проигрывать музыку. В компоненте **Audio**{ } есть обработчик сигнала **onStopped** для сигнала "stopped [остановлен]" компонента **Audio**, который возникает при остановке звукового файла функцией **stop()** или при остановке воспроизведения по достижении конца файла. В обработчике **onStopped** мы проверяем, что состояние воспроизведения True, и если да, меняем цвет кнопки на **UbuntuColors.warmGrey** (один из цветов официальной цветовой палитры Ubuntu), который означает, что звук воспроизводится, а затем вызываем функцию **play()** для воспроизведения звука. Эта функция сможет проигрывать звук в цикле (если состояние было установлено в True, мы знаем, что звук уже проигрывался и его нужно проигрывать в цикле).

Теперь рассмотрим последний компонент, **MouseArea**. Этот компонент создает область, доступную для щелчков, на всей **UbuntuShape**{ } и на каждом из его дочерних компонентов. Здесь мы отвечаем на сигнал «нажато [pressed]» в обработчике

Вливайтесь в разработчики

Дружное общество разработчиков Ubuntu растет и развивается. Начните свое путешествие с <http://developer.ubuntu.com>, и если вам придется там по душе, можете

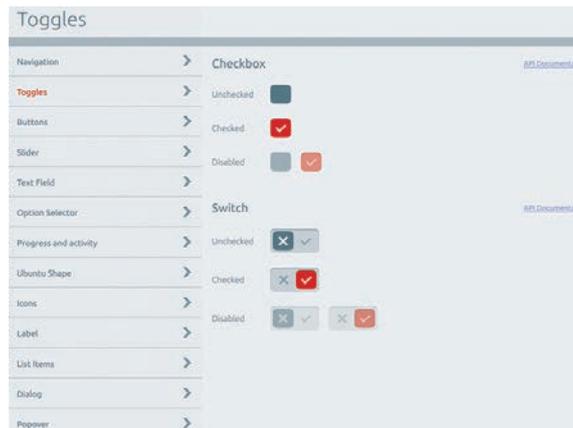
заглянуть в их сообщество Google+ на <http://bit.ly/1bBeYUJ>, а также в сообщество Facebook на www.facebook.com/ubuntuappdev.

onPressed, и проверяем, равна ли переменная состояния воспроизведения False. Если да (звук не проигрывается), мы вызываем функцию **play()**, задаем цвет кнопки **UbuntuShape()** как умеренно серый и устанавливаем переменную в True. В противном случае, если переменная равна True, мы вызываем **stop()** для остановки воспроизведения, устанавливаем соответствующий цвет кнопки и затем устанавливаем переменную в False.

Прежде чем добавить звуковые кнопки в главный интерфейс, создадим каталог "media" в каталоге нашего проекта "sleepy" и добавим туда несколько иконок и звуковых файлов. Я добавил следующие: **bigwaves.png**, **bigwaves.mp3**, **calmwaves.mp3**, **calmwaves.png**, **chimes.png**, **chimes.mp3**, **city.png**, **city.mp3**, **countryside.png**, **countryside.mp3**, **whitenoise.png** и **whitenoise.mp3**. Как видите, имена файлов для PNG и MP3 попарно совпадают. Эти файлы можно скопировать в каталог в обычном файловом менеджере.

Теперь добавим в **sleepy.qml** наши звуковые кнопки. Добавьте такой код внутри компонента Page:

```
Item {
    anchors.fill: parent
    id: wrapper
    property int n_columns: height > width ? 2 : 3;
    property int n_rows: height > width ? 3 : 2;
    property int button_size: Math.min (width / n_columns, height / n_rows) * 0.9;
    property int button_radius: 10;
    property int button_xspacing: (width - button_size * n_columns) / (n_columns + 1);
    property int button_yspacing: (height - button_size * n_rows) / (n_rows + 1);
    Grid {
        x: wrapper.button_xspacing;
        y: wrapper.button_yspacing;
        columns: wrapper.n_columns;
        rows: wrapper.n_rows;
        columnSpacing: wrapper.button_xspacing;
        rowSpacing: wrapper.button_yspacing;
        SoundButton {
            width: wrapper.button_size;
            height: wrapper.button_size;
            radius: wrapper.button_radius;
            color: "#32222C"
            description: "Белый шум"
            imageSource: "media/whitenoise.png";
            soundSource: "media/whitenoise.mp3";
        }
        SoundButton {
            width: wrapper.button_size;
            height: wrapper.button_size;
            radius: wrapper.button_radius;
            color: "#32222C"
            description: "Порывы ветра"
            imageSource: "media/chimes.png";
            soundSource: "media/chimes.mp3";
        }
        SoundButton {
            width: wrapper.button_size;
            height: wrapper.button_size;
            radius: wrapper.button_radius;
            color: "#32222C"
            description: "Бурные волны"
            imageSource: "media/bigwaves.png";
            soundSource: "media/bigwaves.mp3";
        }
        SoundButton {
            width: wrapper.button_size;
```



› Набор компонентов Ubuntu включает весьма обширен, и их можно посмотреть в Витрине компонентов.

```
        height: wrapper.button_size;
        radius: wrapper.button_radius;
        color: "#32222C"
        description: "Тихий плеск волн"
        imageSource: "media/calmwaves.png";
        soundSource: "media/calmwaves.mp3";
    }
    SoundButton {
        width: wrapper.button_size;
        height: wrapper.button_size;
        radius: wrapper.button_radius;
        color: "#32222C"
        description: "Шум большого города"
        imageSource: "media/city.png";
        soundSource: "media/city.mp3";
    }
    SoundButton {
        width: wrapper.button_size;
        height: wrapper.button_size;
        radius: wrapper.button_radius;
        color: "#32222C"
        description: "Деревенская тишина"
        imageSource: "media/countryside.png";
        soundSource: "media/countryside.mp3";
    }
    } } }
```

Здесь мы создаем компонент **Item()** с компонентом **Grid()** внутри него. Прежде всего мы выполняем некоторые вычисления в компоненте **Item()**, чтобы определить размер кнопок на основе количества столбцов (3) и строк (2). Затем в компоненте **Grid()** мы создаем наши звуковые кнопки, которые ссылаются на наш код QML. После этого внутри каждого компонента **SoundButton()** мы передаем описание, которое задает текст метки **Label()** в **SoundButton()**, а также **imageSource** и **soundSource**, которые ссылаются на наши алиасы в компоненте **SoundButton()**, которые в свою очередь устанавливают источники звука и изображения **Image.source** и **Audio.source**.

Теперь, запустив приложение, вы увидите кнопки, сможете проигрывать мелодии и останавливать воспроизведение, и все будет работать как должно. Перед завершением добавим немного глянца: красивый градиент на фоне главного окна **MainView()**. Добавьте такие строки после **height: units.gu(75)** в файл **MainView**:

```
        backgroundColor: "#741266"
        footerColor: "#bd0776"
```

Теперь в приложении появился красивый градиент.

Здесь мы коснулись лишь небольшой части возможностей SDK Ubuntu и исчерпывающего набора инструментов Ubuntu Components и QML. К счастью, на сайте <http://developer.ubuntu.com> гораздо больше материала, там вы сможете найти документацию по API, примеры кода, руководства и многое другое. Удачи! **LXF**



Erlang: Подводим

Андрей Ушаков завершает серию практикумов по многозадачности, выставляя функциям оценки за производительность.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

Данной статьей мы заканчиваем наш практикум по многозадачности, посвященный созданию разнообразных многозадачных реализаций функций **map** и **reduce**. Мы проделали большую работу и создали несколько различных многозадачных реализаций функций **map** и **reduce**. Теперь пришла пора их сравнить. Сравнить мы их в этот раз будем только по одному критерию: производительности.

Прежде чем сравнивать производительность разных реализаций функций **map** и **reduce**, необходимо разработать методику их сравнения. Для измерения времени выполнения некоторой функции мы будем использовать функцию **timer:tc/1** из модуля **timer**. Эта функция принимает в качестве аргумента некоторую функцию **Fun** и возвращает кортеж из двух элементов (**Time**, **Value**), где **Time** — время выполнения функции **Fun** в микросекундах, а **Value** — значение, возвращаемое этой функцией. Но измерять время выполнения некоторой функции при помощи функции **timer:tc/1** не так уж и просто, как может это показаться. Давайте, например, измерим время выполнения функции **lists:seq(1,5)**, которая генерирует список чисел от 1 до 5. Для этого запустим среду времени выполнения **Erlang** и введем в ней следующее выражение: **timer:tc(fun() -> lists:seq(1,5) end)**. Результатом этого выражения будет следующее значение: **{0, [1,2,3,4,5]}**, откуда следует, что время выполнения функции **lists:seq(1,5)** равно 0. На первый взгляд может показаться, что функция **timer:tc/1** работает неправильно, так как в действительности генерация списка **[1,2,3,4,5]** занимает некоторое время, а не происходит мгновенно. Однако если вспомнить, что время на компьютере обновляется при помощи таймера, имеющего определенную разрешающую способность, то все становится на свои места. В действительности оказывается, что время выполнения функции **lists:seq(1,5)** меньше разрешающей способности таймера; именно поэтому мы получаем 0 в качестве значения времени выполнения этой функции с данными аргументами.

Возникает вполне логичный вопрос: как нам измерить время выполнения некоторой быстрой функции? Для понимания этого давайте ответим на другой вопрос: как измерить толщину обычного листа бумаги, причем достаточно точно? Если мы будем решать эту задачу в лоб (т.е. возьмем линейку или штангенциркуль и будем измерять толщину одного листа бумаги), то ответ будет очевиден: никак. Однако если мы возьмем стопку из 1000 листов бумаги, измерим ее толщину и разделим полученное значение

на количество листов бумаги в стопке (в нашем случае на 1000), то мы получим толщину одного листа бумаги; это значение будет достаточно точным. Аналогичным образом мы можем измерить и время некоторой быстрой функции: выполнить эту функцию **N** раз подряд, измерить время выполнения и разделить полученное значение на **N**. И мы получим время выполнения интересующей нас функции, но не так точно, как ожидается. Все дело в том, что параллельно с нами работает такой системный компонент, как сборщик мусора [garbage collector], причем периоды его работы для нас случайны. Пусть во время выполнения некоторой функции **N** раз подряд сборка мусора произошла **M** раз. Пусть время однократного выполнения интересующей нас функции — t_f , а время сборки мусора — t_{gc} (для простоты считаем, что все сеансы сборки мусора занимают одинаковое время). Тогда, используя приведенную выше методику, мы получим следующее значение времени: $(N \cdot t_f + M \cdot t_{gc}) / N = t_f + (M/N) \cdot t_{gc}$, т.е. верхнюю границу для времени выполнения интересующей нас функции. С одной стороны, чем больше значение **N** относительно **M**, тем ближе полученное нами значение к истинному времени выполнения интересующей нас функции. С другой стороны, мы не можем никак влиять на сборку мусора, поэтому вряд ли нам удастся сделать число **N** сильно больше числа **M**. Если быть более точным, то мы можем запустить внеплановую сборку мусора при помощи функции **garbage_collect/0,1**, но не можем отменить запланированную сборку мусора (или совсем прекратить ее на время). Более того, сборка мусора вносит свой вклад во время выполнения всегда: и во время наших измерений, и во время реального выполнения некоторой функции. Поэтому под временем выполнения мы будем понимать время выполнения вместе с возможной сборкой мусора.

Помимо сборки мусора, существует еще множество факторов, влияющих на измерение времени работы некоторой функции. Эти факторы связаны с работой компьютерного «железа», операционной системы и среды времени выполнения Erlang. Так, например, выполнение интересующей нас функции на процессоре (или на ядре процессора) может быть вытеснено более приоритетной задачей (при выполнении на операционной системе с вытесняющей многозадачностью). Все эти факторы влияют на измерение времени работы некоторой интересующей нас функции, причем (как и в случае со сборкой мусора) в сторону увеличения измеренного значения. Погрешности, вносимые этими факторами, носят как систематический (потому что происходят периодически), так и случайный (потому что невозможно предсказать, когда эти факторы в следующий раз повлияют на наши измерения) характер. Систематическую часть погрешности мы не будем трогать; она вносит одинаковый вклад как в процесс измерения, так и в реальную работу некоторой функции. Просто под временем выполнения мы будем понимать время выполнения вместе со всеми «паразитными» вкладами (от сборщика мусора, от планировщика заданий и т.д.). А вот значение случайной погрешности мы можем уменьшить. Для этого нам нужно провести измерение интересующей нас величины несколько раз и вычислить ее среднее значение. Чем больше будет количество измерений интересующей нас

Систематическая погрешность

Систематическая погрешность — это погрешность, изменяющаяся во времени по определенному закону (частным случаем является постоянная погрешность, не изменяющаяся с течением времени). Систематические погрешности могут быть связаны с ошибками приборов

(неправильная шкала, калибровка и т.п.), не учтенными экспериментатором.

Систематическую погрешность нельзя устранить повторными измерениями. Ее устраняют либо с помощью поправок, либо посредством «улучшения» эксперимента (повышения его чистоты).

ИТОГИ

величины, тем точнее будет эта величина и тем меньше будет значение случайной части погрешности.

Давайте подведем промежуточный итог по тому, как мы будем измерять время выполнения интересующей нас функции. Во-первых, мы будем измерять время выполнения «пакета», состоящего из нескольких вызовов интересующей нас функции. Время выполнения интересующей нас функции мы получим, разделив время выполнения «пакета» на количество вызовов в «пакете». Во-вторых, вычислять время выполнения интересующей нас функции мы будем несколько раз, для уменьшения случайной части погрешности вычисления. По полученному набору значений времени выполнения мы будем вычислять среднее значение и среднеквадратичное отклонение (СКО). Среднеквадратичное отклонение поможет оценить нам случайную часть погрешности. Давайте эту методику измерения времени выполнения интересующей нас функции мы реализуем. Все функции, относящиеся к измерению времени выполнения, мы расположим в отдельном модуле `performance_tester`. А начнем мы с «пакетов», состоящих из нескольких вызовов интересующей нас функции. Функция `time_test_body/2` (внутренняя относительно модуля `performance_tester`) служит для выполнения «пакета» из вызовов интересующей нас функции `Fun`:

```
time_test_body(_Fun, 0) -> true;
time_test_body(Fun, TestCount) ->
  Fun(),
  time_test_body(Fun, TestCount - 1).
```

Функция `time_test/2` (внутренняя относительно модуля `performance_tester`) служит для формирования «пакета» вызовов, его выполнения и вычисления времени однократного выполнения интересующей нас функции `Fun`:

```
time_test(Fun, TestCount) ->
  {Time, _Value} = timer:tc(fun() -> time_test_body(Fun,
  TestCount) end),
  Time / TestCount.
```

В этой функции для измерения времени выполнения «пакета» мы используем функцию `timer:tc/1`. Следующая функция, которую мы рассмотрим — функция `time_test/4` для получения нескольких значений времени выполнения интересующей нас функции:

```
time_test(_Fun, _TestCount, 0, Values) -> Values;
time_test(Fun, TestCount, Count, Values) ->
  Time = time_test(Fun, TestCount),
  erlang:garbage_collect(),
  time_test(Fun, TestCount, Count - 1, [Time] ++ Values).
```

В реализации этой функции интересно обратить внимание на следующее: на вызов функции `erlang:garbage_collect/0` между измерениями значений времени выполнения функции. Это нужно для того, чтобы убрать весь мусор, возникший после выполнения «пакета». После того, как мы получаем несколько значений времени выполнения, нам необходимо вычислить среднее значение этого времени и среднеквадратичное отклонение этого среднего. Для этого мы создаем две следующие функции — `calc_mean/1` и `calc_standard_deviation/2` (также внутренние функции модуля `performance_tester`):

Случайная погрешность

Случайная погрешность — это составляющая погрешности измерения, изменяющаяся случайным образом в серии повторных измерений одной и той же величины, проведенных в одних и тех же условиях. В появлении таких погрешностей не наблюдается какой-либо закономерности, они обнаруживаются при повторных измерениях одной и той же величины в виде некоторого разброса получаемых результатов. Случайные погрешности неизбежны, неустранимы и всегда присутствуют в результате измерения, однако их влияние, как

правило, можно снизить статистической обработкой. Описание случайных погрешностей возможно только на основе теории случайных процессов и математической статистики. Основным свойством случайной погрешности является возможность уменьшения искажения искомой величины путем усреднения данных. Уточнение оценки искомой величины при увеличении количества измерений (повторных экспериментов) означает, что среднее случайной погрешности при увеличении объема данных стремится к 0 (закон больших чисел).

```
calc_mean(Values) ->
  lists:sum(Values) / length(Values).
calc_standard_deviation(Mean, Values) ->
  Sum = lists:foldl(fun (Value, Acc) -> Acc + (Value - Mean) *
  (Value - Mean) end, 0, Values),
  math:sqrt(Sum / length(Values)).
```

Функция `calc_mean/1` вычисляет среднее значение по набору значений; функция `calc_standard_deviation/2` вычисляет среднеквадратичное отклонение по набору значений и вычисленному среднему значению для этого набора. И, наконец, функция `performance_tester:time_test/3` является точкой входа (и экспортируемой функцией из модуля `performance_tester`) для измерения времени выполнения:

```
time_test(Fun, TestCount, Count) ->
  TimeValues = time_test(Fun, TestCount, Count, []),
  Mean = calc_mean(TimeValues),
  Deviation = calc_standard_deviation(Mean, TimeValues),
  erlang:garbage_collect(),
  {Mean, Deviation}.
```

В эту функцию мы передаем функцию `Fun`, время выполнения которой мы хотим измерить, количество вызовов функции `Fun` в «пакете» `TestCount` и количество значений в наборе `Count` (для вычисления среднего и среднеквадратичного отклонения). По полученному набору значений времени вычисления мы вычисляем среднее значение, среднеквадратичное отклонение и возвращаем полученные результаты в виде кортежа из двух значений.

Договоримся о методике сравнения производительности разных реализаций функций `map` и `reduce`. В качестве исходных данных для реализаций функций `map` и `reduce` мы будем использовать список из целых чисел. Для минимизации размера этого списка следует использовать целые числа минимального размера. Минимальный размер целых чисел — 1 слово; на 32-битных системах целое число занимает 1 слово при условии, что его значение лежит в диапазоне от `-134217729` до `134217728`. Кроме того, для большей чистоты получаемых результатов, мы будем использовать не предопределенный список целых чисел, а список чисел, »

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Таблица 1

N	1	2	3	4a	4б	5	6	7
1000	2020 (30)	1990 (20)	4040 (30)	1110 (20)	1120 (30)	1130 (20)	2810 (50)	1140 (20)
10000	26000 (3000)	26000 (3000)	42400 (500)	6900 (200)	6900 (200)	6900 (300)	8900 (400)	7000 (200)
100000	360000 (30000)	358000 (3000)	520000 (7000)	71000 (2000)	71000 (2000)	84000 (6000)	51000 (2000)	90000 (4000)
1000000	4200000 (400000)	4200000 (300000)	5500000 (200000)	870000 (30000)	870000 (30000)	850000 (40000)	580000 (10000)	900000 (10000)

генерируемых случайным образом. Для этого мы будем использовать функцию `data_generator:generate_int_data/2`, определенную в модуле `data_generator` (см. исходные коды на диске). При построении такой зависимости мы будем проводить измерения при следующих размерах списка исходных данных: 1000, 10000, 100000, 1000000. При измерении производительности разных реализаций функции `map` в качестве функции отображения мы будем использовать функцию `fun math:sqrt/1`. При измерении производительности разных реализаций функции `reduce` в качестве функции свертки мы будем использовать функцию `fun(Number, Sum) -> Sum + math:sqrt(Number) end`; в качестве начального значения операции свертки мы будем брать 0.

Теперь давайте поговорим про получение результатов. Получать результаты производительности мы будем следующим образом: для каждой реализации функции `map` или `reduce` и для каждого размера исходного списка мы будем создавать новый экземпляр среды Erlang (или несколько, если необходимо). После получения результата мы будем завершать работу созданного экземпляра среды Erlang при помощи вызова функции (BIF) `halt/0`.

Последний шаг, который необходимо сделать перед тем, как переходить непосредственно к данным, это привести данные о конфигурации используемого оборудования. Для получения результатов использовали два компьютера со следующей конфигурацией: процессор Intel Core i5-2400 (с частотой 3,10 ГГц), память 8 ГБ ОЗУ, операционная система Ubuntu 13.04 64-бит, пропускная способность сети между компьютерами 100 Мбит/с.

Перейдем к полученным результатам. Таблица 1 содержит время выполнения разных реализаций функции `map` в зависимости от размера списка исходных данных.

Здесь столбец N содержит количество элементов (чисел) в списке исходных данных. Столбец 1 содержит время выполнения стандартной реализации: функции `lists:map/2`. Столбец 2 содержит время выполнения функции `parallel_map:usual_map/2`. Столбец 3 содержит время выполнения функции `parallel_map:simple_pmap/2`. Столбцы 4a и 4б содержат время выполнения функций `parallel_map:portion_pmap/3` и `parallel_map:portion_gen_pmap/3` соответственно. Столбец 5 содержит время выполнения функции `parallel_map:limited_pmap/4`. Столбец 6 содержит время выполнения функции `parallel_map:distributed_pmap/5`. И, наконец, столбец 7 содержит время выполнения функции `parallel_map:smartmsg_pmap/4`. Размеры порций данных (для данных из столбцов 4–7) устанавливаются следующим образом: 250 элементов при размере исходных данных 1000 чисел, 2500 элементов при размере исходных данных 10000 чисел и 10000 элементов во всех остальных случаях. При получении данных из столбцов 5 и 7 мы использовали 4 рабочих процесса. При получении данных из столбца 6 мы использовали 2 узла с 2 рабочими процессами на каждом из них.

В таблице 2 столбец N содержит количество элементов (чисел) в списке исходных данных. Столбец 1 содержит время выполнения стандартной реализации: функции `lists:foldl/3`. Столбец 2 содержит время выполнения функции `parallel_reduce:usual_reduce/2`. Столбцы 3a и 3б содержат время выполнения функций `parallel_reduce:portion_reduce/4` и `parallel_reduce:portion_gen_reduce/4` соответственно. Столбец 4 содержит время выполнения функции `parallel_reduce:limited_reduce/5`. Столбец 5 содержит время выполнения функции `parallel_reduce:distributed_reduce/6`. И, наконец, столбец 6 содержит время выполнения функции `parallel_reduce:smartmsg_reduce/5`. Размеры порций данных (для данных из столбцов 3–6) устанавливаются следующим образом: 250 элементов при размере исходных данных 1000 чисел, 2500 элементов при размере исходных данных 10000 чисел и 10000 элементов во всех остальных случаях. При получении данных из столбцов 4 и 6 мы использовали 4 рабочих процесса. При получении данных из столбца 5 мы использовали 2 узла с 2 рабочими процессами на каждом из них. Все значения в обеих таблицах приведены в микросекундах.

Теперь мы можем перейти к цели этой статьи: к сравнению производительности разных реализаций функций `map` и `reduce` на основе приведенных выше данных. Из приведенных выше данных и условий тестирования производительности можно сделать следующие выводы:

- 1 Для всех измеренных данных о времени выполнения мы посчитали среднеквадратичное отклонение (СКО). Значение СКО приведено в скобках за соответствующим значением времени выполнения в той же ячейке таблицы. Для всех измеренных данных значение СКО не больше 10% от соответствующего значения времени выполнения. Это означает, что случайная погрешность находится в допустимых рамках. Что же касается систематической погрешности, то факторы, которые влияют на значение систематической погрешности в наших измерениях, точно так же влияют и на время выполнения в реальной жизни. Поэтому мы можем считать, что полученное нами значение времени выполнения близко к своему значению при реальной работе.
- 2 Стандартная реализация `lists:map/2` и обычная, не многозадачная реализация `parallel_map:usual_map/2` функции `map` являются наименее эффективными (за одним единственным исключением), что ожидаемо. Это означает, что их выполнение занимает наибольшее время по сравнению со всеми другими реализациями. При этом стандартная реализация `lists:map/2` и обычная, не многозадачная реализация `parallel_map:usual_map/2` функции `map` одинаково эффективны. То же самое справедливо и для реализаций `lists:foldl/3` и `parallel_reduce:usual_reduce/2` функции `reduce`.
- 3 Никогда, никогда, никогда не используйте реализацию `parallel_map:simple_pmap/2` функции `map`. Это приведет

Таблица 2

N	1	2	3a	3б	4	5	6
1000	3810 (40)	3720 (20)	1810 (30)	1800 (30)	1860 (20)	2440 (60)	1850 (20)
10000	38900 (100)	38300 (300)	11000 (100)	11000 (200)	11000 (100)	5600 (300)	10800 (200)
100000	383000 (2000)	385000 (3000)	104000 (3000)	106000 (2000)	122000 (3000)	28000 (1000)	117000 (2000)
1000000	3850000 (10000)	3890000 (30000)	1090000 (2000)	1090000 (4000)	1230000 (9000)	330000 (5000)	1040000 (20000)

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

к преждевременной «пессимизации» производительности, т. к. время выполнения этой функции сильно больше, чем время выполнения обычных не многозадачных реализаций.

4 Реализация на основе порций данных `parallel_map:portion_pmap/3` и обобщенная реализация на основе порций данных `parallel_map:portion_gen_pmap/3` функции `map` имеют одинаковую производительность. Это означает, что переход от обычной реализации на основе порций к обобщенной реализации не вносит никаких дополнительных затрат ко времени выполнения. Все наши дальнейшие реализации также основываются на обобщенном подходе. Поэтому мы можем быть уверены, что данный подход не вносит никаких дополнительных затрат ко времени выполнения ни для какой из реализаций. То же самое справедливо и для всех реализаций на основе порций функции `reduce`.

5 Сравнение реализации на основе порций `parallel_map:portion_gen_pmap/3` с реализацией с ограничениями на количество рабочих процессов `parallel_map:limited_pmap/4` функции `map` на самом деле не тривиально. Если посмотреть на приведенные данные, то может показаться, что реализация `parallel_map:portion_gen_pmap/3` эффективнее, чем `parallel_map:limited_pmap/4`. Но не стоит забывать, что при получении времени выполнения для реализации `parallel_map:limited_pmap/4` мы использовали 4 рабочих процесса (по числу ядер на компьютере). Подбор необходимого количества рабочих процессов для минимизации времени выполнения реализации `parallel_map:limited_pmap/4` следует для каждой задачи и конфигурации компьютера выполнять индивидуально. Так, например, для исходного списка из 1 000 000 элементов и размера порции в 100 элементов время выполнения реализации `parallel_map:portion_gen_pmap/3` будет 10,81 секунд. Время выполнения реализации `parallel_map:limited_pmap/4` с 4 рабочими процессами будет 10,97 секунд, а время выполнения реализации `parallel_map:limited_pmap/4` с 64 рабочими процессами будет 10,67 секунд. Все это справедливо и для реализаций `parallel_reduce:portion_gen_reduce/4` и `parallel_reduce:limited_reduce/5` функции `reduce`.

6 Реализация функции `map` с созданием заданий по мере необходимости `parallel_map:smartmsg_pmap/4` менее эффективна, чем реализация с созданием всех заданий сразу `parallel_map:limited_pmap/4`. Однако разница в производительности достаточно незначительна (например, при размере исходного списка в 100 000 элементов разница в производительности будет менее 6%). Поэтому, если нет жестких требований к производительности, которыми реализация `parallel_map:smartmsg_pmap/4` не удовлетворяет (а это означает, что реализация `parallel_map:limited_pmap/4` удовлетворяет этим требованиям впритык, без особого запаса), то лучше использовать ее. Причина этого в том, что реализация `parallel_map:smartmsg_pmap/4` не создает таких пиковых нагрузок на память и сеть, как реализация `parallel_map:limited_pmap/4`. Все это справедливо и для реализаций `parallel_reduce:smartmsg_reduce/5` и `parallel_reduce:limited_reduce/5` функции `reduce`.

7 Распределенная реализация `parallel_map:distributed_pmap/5` функции `map` гораздо эффективнее всех остальных реализаций для списков исходных данных большего размера. Очевидно, в этом случае выгоды от распределения вычислений на нескольких компьютерах превышают затраты на передачу данных по сети. То же самое справедливо и для реализации `parallel_reduce:distributed_reduce/6` функции `reduce`.

8 Данные о производительности реализаций функции `map` на основе пула узлов `parallel_map:pool_pmap/5` и `parallel_map:pool_pmap_impl/4` не присутствуют в нашей таблице. Связано это с ошибкой реализации самого пула узлов в модуле `pool`: функция `pool:pspawn_link/3` создание процесса и создание связи с этим процессом делает не атомарно. И проявляется эта ошибка при попытке использовать пул узлов с достаточно быстрыми задачами, как у нас. То же самое справедливо и для реализаций `parallel_`

Среднеквадратичное отклонение

Среднеквадратичное отклонение — в теории вероятностей и статистике наиболее распространенный показатель рассеивания значений случайной величины относительно ее математического ожидания. Измеряется в единицах измерения самой случайной величины. Равно корню квадратному из дисперсии случайной величины:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - m)^2}$$

Здесь σ — среднеквадратичное отклонение, n — объем выборки, x_i — i -й элемент выборки, m — среднее арифметическое выборки.

`reduce:pool_reduce/6` и `parallel_reduce:pool_reduce_impl/5` функции `reduce`.

9 И, наконец, давайте скажем пару слов о плате за повышение эффективности. Любой многозадачный алгоритм и любая многозадачная реализация обычно сложнее соответствующих однозадачных алгоритмов и реализаций. Кроме того, и объем кода многозадачных реализаций обычно больше. Так, например, однозадачная реализация `parallel_map:usual_map/2` содержит всего одно выражение, тогда как многозадачные реализации функции `map` существенно больше. Это означает, что сложность тестирования, сопровождения, дальнейшего развития многозадачных реализаций больше, чем соответствующих однозадачных реализаций.

Давайте скажем пару слов и об определении потребляемой памяти. Сделать это можно следующим образом: вычислить количество используемой памяти до и после вызова (например, при помощи функции `erlang.memory/1` с параметром `processes_used`), после чего получить их разность. Проблема в том, что полученное значение может быть любым и никак не связанным с реальным потреблением памяти. Причина этого в сборщике мусора. Поэтому более правильным было бы определение пикового потребления памяти в процессе выполнения функции, а также зависимости потребления памяти во времени. Получение подобных данных — это тема отдельной статьи; в будущем мы вернемся к этому вопросу.

Вот мы и закончили наш практикум по многозадачности. Но это не означает, что мы закончили вообще разговор про многозадачность: к этой теме мы еще не раз вернемся. Точно так же мы не раз еще вернемся и к этой задаче: к созданию разнообразных реализаций функций `map` и `reduce`, конечно, в рамках нескольких других вопросов. А в следующий раз мы начнем разговор о тестировании программного обеспечения самим разработчиком. LXF

Хронометраж для функций

Время выполнения интересующей нас функции в языке Erlang можно измерить одним из следующих способов:

- » Использовать семейство функций `timer:tc/1,2,3` из модуля `timer` (что мы и делаем в нашей статье).
- » Измерить время до и после выполнения интересующей нас функции при помощи функции (BIF) `now/0`, после чего посчитать разность полученных значений (например, при помощи функции `timer:now_diff/2` из модуля `timer`).

- » Измерить время до и после выполнения интересующей нас функции при помощи функции `os:timestamp/0` из модуля `os`, после чего посчитать разность полученных значений (например, при помощи функции `timer:now_diff/2` из модуля `timer`).
- » Измерить время до и после выполнения интересующей нас функции при помощи функции (BIF) `statistics/1`, после чего посчитать разность полученных значений.

Blender: Внедريم

Андрей Прахов вознамерился воспользоваться возможностями Motion Tracking на полную мощь! Сделаем новый шаг от реальности к виртуальности.



Наш эксперт

Андрей Прахов
Участник нескольких игровых проектов, представитель СМИ, автор первой русской книги по *Blender*: «Blender. 3D-моделирование и анимация. Руководство для начинающих».



Джедаи, бьющиеся на лазерных мечах, смертельные жаля, вылетающие из рук героев *Mortal Combat*, ангелы и демоны в фильме «Константин»... современное кино изобилует потрясающими эффектами. Более-менее подкованный зритель махнет рукой, да и скажет: «Трехмерная графика!» И будет совершенно прав. Минули те времена, когда

спецеффекты в кино создавались хитроумной операторской работой, трудом костюмеров, гримеров и каскадеров.

Современные технологии

Возьмем, к примеру, известный фильм «Матрица». Помните сцену, когда главный герой уклонялся от выстрелов? Казалось, время в этот момент замедлилось, пули буравили воздух, оставляя светящийся след. Нео ловко уворачивался, изгибаясь назад под невыслышимым углом, а эффектная работа камеры показывала самые драматичные ракурсы. Действительно впечатляющая работа, но как?..

Посмотрите на рис. 1, где фотокамера запечатлела работу съемочной группы фильма «Матрица». Что такое «green screen», вы уже знаете. В прошлой статье как раз рассматривалась работа с виртуальными задниками в *Blender*. Если вкратце, то технология использования однотонных задников [keying] во время съемок позволяет программно заменить фоновый цвет на статичное или движущееся изображение. Проще говоря, цвет задника становится прозрачным. Вы уже попытались мысленно подставить вместо зеленого цвета вокруг Нео пейзаж сюрреалистического города? Подобную замену можно прекрасно сделать и в *Blender*.

В батальных сценах «Матрицы» очень часто использовались эффектные облеты камеры вокруг застывшего героя. Технически это выполнялось путем моментальной съемки множеством фотоаппаратов, расположенных вокруг Киану Ривза [Keanu Reeves] (на рис. 1 отчетливо видны окуляры фотоаппаратов, встроенных в зеленый бордюр). Эта технология получила название «bullet time» (зарегистрированный товарный знак Warner Bros). Смысл ее в том, что полученные снимки склеивались в последовательность кадров и таким образом имитировалось движение камеры при «остановленной» картинке.

Теперь давайте представим, что кроме фонового города, нужно добавить в сцену летящие пули с эффектом воздушного следа. Если последнее выполняется путем использования частиц, отдельных мешей и материалов с рефракцией, то с анимацией появляются определенные сложности. Достаточно просто «выстрелить» пулей в неподвижной сцене, но учитывая круговой поворот «bullet time» камеры, здесь придется столкнуться с точной синхронизацией 3D с видео. Вручную такое сделать крайне сложно. Ведь виртуальная камера должна в точности повторить траекторию и движение реальной! Вот тут на помощь приходят технологии Motion Tracking.

Охота за артефактами

В любой картинке можно найти места, где изображение является контрастным к близлежащему фону. Например: белый камешек на сером асфальте, зеленый листок на фоне коры, буквы на фасаде здания. Представьте, что программа «знает» эти метки и умеет их отслеживать. Теперь появляется возможность реконструировать движения реальной камеры и перенести их на виртуальную. Именно это является задачей Motion Tracking.

Вернемся опять к съемкам фильма «Матрица». Внимательно присмотревшись к рис. 1, вы заметите кресты, расположенные



» Рис. 1. Эпизод со съемочной площадки фильма «Матрица» (<http://sabia.tic.udc.es/>).

В ТЕЛЕВИДЕНИЕ



► Рис. 2, а. Так было при съемке фильма «Живая сталь» (www.fxguide.com).



► Рис. 2, б. А вот таким получился окончательный результат: был человек — стал робот...

на зеленых стенах задника. Они не сильно контрастируют с основным фоном, но этого достаточно, чтобы программа трекинга могла отследить их при перемещении. А уж скрыть метки при композитинге — дело несложное.

Но это еще не все! С помощью данной технологии удастся отследить перемещение конкретного предмета в кадре.

Смотрели фильм «Живая сталь [Real Steel]» режиссера Шона Леви [Shawn Levy]? Это яркий образец того, как с помощью Motion Tracking можно заменять одни объекты в съемке другими. Так, на рис. 2, а актер, разговаривающий с Хью Джекманом [Hugh Jackman], после обработки предстает в образе могучего робота. Странные же цветные пятна на его костюме — не что иное, как метки для трекинга (рис. 2, б).

Работа с Motion Tracking — привилегия совсем небольшого количества специализированных программ, и, к нашему счастью, в этом списке присутствует *Blender*. По сути, эта технология — часть системы композитинга. Ведь для достижения результата необходимо не только «следить» за метками, но и синхронизировать их с объектами, работать с виртуальными задниками, да и банально скрывать уже ненужные метки в готовом видео.

Для всего этого *Blender* предлагает специальный редактор Movie Clip Editor. А так как популярность работы с технологиями Motion Tracking только нарастает, то разработчики программы

«Работа с Motion Tracking начинается с правильной съемки видео.»

создали специальную заготовку [screen lay-out] с одноименным названием (рис. 3).

Работа с Motion Tracking, как бы банально это ни звучало, начинается с правильной съемки видео. Нужно заранее позаботиться о достаточном количестве меток в каждом кадре. Ведь одним из условий корректного построения траекторий движения объектов является наличие не менее 8 точек. Такое количество — это требование *Blender*, и для других программ оно может отличаться.

Метки могут быть естественными (контрастные участки) и искусственными. Правда, в последнем случае их придется перекрывать в готовом видео. Само расположение ключей зависит от поставленной задачи. Допустим, если нужно виртуализировать камеру, то метки должны

быть расположены как по горизонтали, так и по вертикали. В дальнейшем одну часть нужно будет ориентировать на поверхность, а другую «на стену». А вот для виртуализации

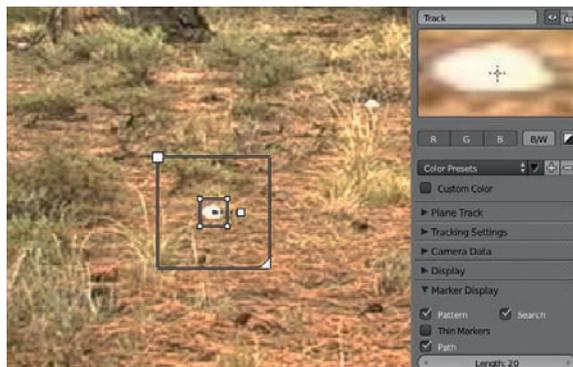
объекта метки должны быть нанесены непосредственно на него (см. рис. 2, а).

Отснятое видео загружается в редактор Movie Clip Editor в виде последовательности кадров или цельного видеофайла (кнопка Open на панели редактора). В первом случае, графические файлы должны иметь пронумерованные названия.

Movie Clip Editor — это не VSE (Video Sequence Editor). У него нет дорожек, при его посредстве нельзя отредактировать или



► Рис. 3. Для Motion Tracking имеется специальная заготовка расположения окон.



► Рис. 4. Белый камушек хорошо контрастирует с пожелтой осенней травой и идеально подходит в качестве метки.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Если Blender «сбился с пути»...

На рис. 4 вокруг выбранного объекта фигурируют две рамки. По умолчанию, при создании метки вам будет показана только одна, а внешнюю нужно включать дополнительно. Конечно, можно обойтись и без нее, но в некоторых случаях она очень полезна.

Чтобы выполнить трекинг указанной метки, программа сканирует определенную область вокруг нее. Именно ее очерчивает внешняя рамка. Предположим, что оператор выполнил резкий проезд камерой и ваша метка в определенный момент

смазалась. В такой момент *Blender* может «сбиться с пути» и проложить неверный маршрут, а то и вовсе отказаться это сделать. Включите рамку в настройках *Marker Display* (пункт *Search*), как на рис. 4, и поиграйте с ее размерами.

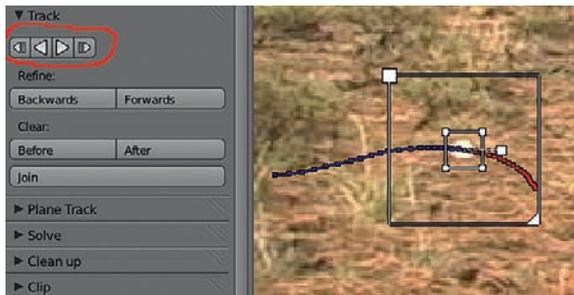


Рис. 5. Настройки трекинга.

смикшировать *strip*. Собственно, стрипов и не подразумевается. *Movie Clip Editor* работает исключительно только с одним загруженным видео. В его функции входят просмотр материала, настройка меток и полная трассировка движений.

Создание и трекинг меток в *Blender* не представляет никакой сложности. Программе достаточно указать в качестве образца нужный участок видео, и она проложит соответствующий маршрут. Правда, от качества метки зависит качество трекинга. В некоторых случаях понадобится ручная подгонка траектории.

Посмотрите на рис. 4, где изображен кусок видео с установленной меткой.

Чтобы установить новый маркер, достаточно щелкнуть левой кнопкой мыши в нужной части видео (удерживая нажатой клавишу *Ctrl*). При этом на экране появится прямоугольная выделенная область, а окно предварительного просмотра справа (см. рис. 4), покажет, что именно вы выделили. Установленный маркер является стандартным объектом *Blender* и подчиняется тем же правилам манипулирования. Его можно перемещать, вращать или изменять форму. Для последнего действия служат точки по углам рамки. Главной задачей на этом этапе является настройка маркера так, чтобы в окне предварительного просмотра умещался сам объект и некоторая контрастирующая с ним область (см. рис. 4). После установки маркера необходимо выполнить трекинг, то есть определить маршрут перемещения. В окне *Movie Clip Editor* слева расположена панель с необходимыми инструментами (рис. 5).

Видите на рис. 5 кнопочки «движения» в панели *Track* (они отмечены красным цветом)? Именно они выполняют трекинг выделенного маркера в указанном направлении от активного кадра сцены. Стоит нажать кнопку-треугольник «вперед», и программа начнет просчитывать маршрут, который

отобразится на экране в виде цветной кривой. Основной сложностью на данном этапе работы с *Motion Tracking* является возможность «проморгать» уход трекинга с реального маршрута движения объекта-метки. Это чревато в дальнейшем неправильным поведением виртуальных объектов.

Чтобы такого не случилось, нужно по кадрам просмотреть предложенный программой маршрут и, при необходимости, скорректировать его. Допустим, если маркер неожиданно «ушел» с указанного объекта, то его нужно вручную переместить на нужное место.

Иногда бывает, что программа прерывает выполнение просчета маршрута, не дойдя до конца видео, или с определенного кадра абсолютно неправильно выполняет трекинг. В этом случае нужно найти то место, где *Blender* потерял указанную метку, и удалить неверный трекинг. Для этого имеются специальные кнопки на панели *Clear* [Очистить], с характерными названиями *Before* и *After*. В первом случае удаляется трекинг до остановленного кадра, а во втором — после (см. рис. 5).

Алгоритм ручной корректировки следующий:

- 1 Находите место сбоя трекинга.
- 2 Удаляете ненужную часть кнопками *After* или *Before*.
- 3 Перемещаете на кадр вперед (или назад) и передвигаете маркер на нужный объект в видео. При этом вы увидите, что программа включит данное перемещение в кривую трекинга.

После всех трудов по настройке маркеров и просчетов трекинга, наступает следующий этап — анимация виртуальной камеры. В левой панели *Movie Clip Editor* имеется закладка с названием *Solve*, а в ней две кнопки: *Camera Motion* и *Clear Solution*. Достаточно нажать первую кнопку, и программа выполнит привязку анимации маркеров к активной камере *Blender*. Если что-то пошло не так, то *Clear Solution* выполнит очистку анимации (рис. 6).

Однако если вы переключитесь в режим просмотра *3D View* с целью увидеть долгожданное движение камеры, то будьте готовы к разочарованию — вы увидите пустую сцену. Для полноценной работы системы *Motion Tracking* понадобится сделать еще немало, и в первую очередь установить прототип сцены. В настройках закладки *Clip* окна *Movie Clip Editor* имеется кнопка с соответствующим названием *Setup Tracking Scene*. После ее нажатия в сцену будет добавлен примитив *Plane* [плоскость]. Именно он является своего рода виртуальной поверхностью для трехмерных объектов.

Вот теперь можно в окне *3D View*, с позиции камеры, увидеть долгожданную привязку 3D к двумерному изображению. Так, если попробовать запустить анимацию, то камера *Blender* должна двигаться точно так же, как и реальная камера оператора. При этом *Blender* выполнит еще одну полезную вещь, а именно — установит текущее видео как бэкграунд к трехмерной сцене.

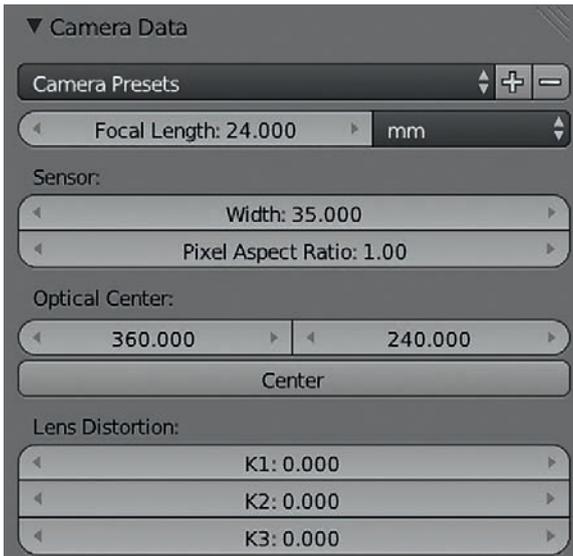
Фантастические превращения

Казалось бы, все. Наполняя сцену нужными объектами, анимируя их и просчитывая; но не тут-то было. Любой добавленный

Рис. 6. Кнопки *Camera Motion* и *Setup Tracking Scene* завершают установку сцены *Motion Tracking*.



» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



► Рис. 7. Дополнительные настройки для камеры.

в сцену фоновый объект отказывается двигаться в соответствии с трекингом, да и в результате рендера отсутствует исходное видео. В первом случае необходима более тонкая настройка трекинга, а во втором поможет всемогущий редактор нод.

Система Motion Tracking в *Blender* имеет массу настроек, скрытых во всевозможных закладках окна редактора. Обычно нет необходимости их изменять, по умолчанию все и так приемлемо работает, но знать о них нужно.

При проблемах с выполнением трекинга меток можно попробовать привести параметры виртуальной камеры в соответствие с реальной, которой проводилась съемка. Для этого в правой панели окна *Movie Clip Editor* предусмотрена закладка *Camera Data* (рис. 7).

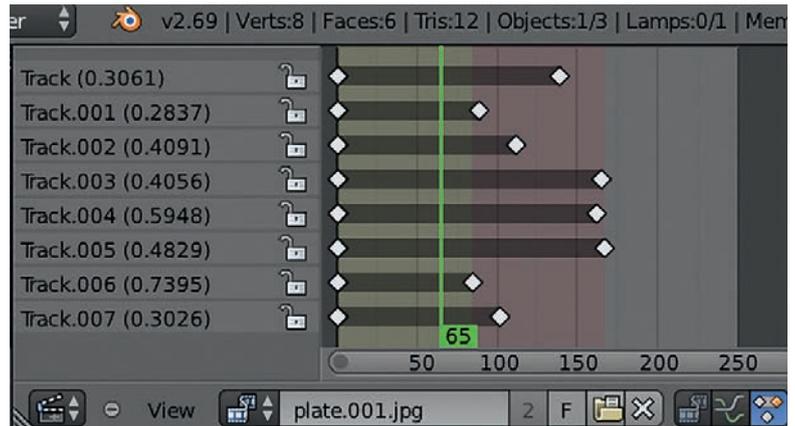
Первоначально стоит поискать название своей камеры в меню пресетов, где собраны наиболее популярные бренды. В случае отсутствия нужного названия можно настроить параметры вручную — это фокусное расстояние, калибровка оптики и т. д. Только не забудьте, что данные настройки выполняются персонально для системы трекинга. Желательно те же настройки установить и для стандартной камеры *Blender* в окне *3D View*.

После запуска просчета кнопкой *Camera Motion* в информационном окне *Blender* может появиться сообщение о наличии некоторых ошибок. При этом обратите внимание на дополнительное окно с названиями всех маркеров сцены (рис. 8).

В скобках программа демонстрирует цифровой показатель качества проекции в 3D-пространстве. Если значение больше 0.3, то при проецировании данного трека возникли некоторые неточности. Чем больше будет «битых» треков, тем хуже результат в целом. В этом случае поможет опция *Refine* в левой части окна (см. рис. 6). В выпадающем меню можно выбрать подходящий способ коррекции ошибок, который в свою очередь зависит от пресета выбранной камеры. В большинстве случаев подходит пункт *K1, K2* меню *Refine*.

Учтите, что любые изменения в настройках трекинга требуют перерасчета проекции. Просто нажмите кнопку *Clear Solution* для очистки, а затем *Camera Motion* для просчета (см. рис. 6).

Еще более точных результатов привязки фонового 3D к видеоизображению можно добиться при использовании возможностей *Reconstruction*.



Окно *Movie Clip Editor* способно функционировать в нескольких режимах. С одним вы уже знакомы: он установлен по умолчанию и называется *Tracking*. В меню *Mode* можно выбрать любой другой (рис. 9).

Режим *Reconstruction* позволяет более точно совместить двумерные маркеры в изображении с их положением в 3D-пространстве.

После выбора этого режима в левой части окна появятся новые опции. Среди них имеется закладка *Orientation*. Ориентация маркеров выполняется по двум направлениям: поверхности пола и стены. Причем это делается не для всех меток в сцене, а только для определенного количества.

Принцип работы следующий:

- 1 Выбрать три метки, визуально расположенные в нижней части видео (пример — дорожное покрытие). Нажать кнопку *Floor* в закладке *Orientation*.
- 2 Выбрать следующие три метки, но уже расположенные в вертикальной плоскости (например, стена дома). Нажать кнопку *Wall* в закладке *Orientation*.

После этих действий трехмерная камера должна быть точно ориентирована в соответствии с реальной камерой. Если этого не произошло, всегда можно сбросить ориентацию маркеров, посредством нажатия кнопки *Set Origin* в закладке *Orientation*.

Как видите, в *Blender* достаточно просто выполнить привязку фоновых трехмерных объектов к реальной камере. Но это еще не все возможности системы *Motion Tracking!* LXF

► Рис. 8. В этом окне показываются результаты трекинга.



► Рис. 9. Выбор режима работы системы трекинга.

LiVES: Интерфейс

Сергей Яремчук продолжает разговор о видеоредакторе *LiVES*, начатый в *LXF175*. Сегодня он расскажет об особенностях работы в режиме Multitrack.



Наш эксперт

Сергей Яремчук
Фрилансер, автор более 1000 статей и 6 книг. 14 лет изучает Linux и пишет статьи, чтобы не забыть пройденное.



Видеоредактор *LiVES* по умолчанию загружается в режиме фрагментарного редактирования — Clip Editor, когда в интерфейсе доступна только одна дорожка. Он удобен для относительно простых проектов (обрезать видео, склеить клип, применить эффекты); также мы получаем весьма продвинутый видеопроигрыватель. В сложных проектах без нескольких дорожек управиться будет проблематично, а значит, знакомства с режимом Multitrack нам не избежать. Правда, есть один отрицательный момент: в этом режиме программа менее стабильна и нередко вылетает, особенно когда пользователь не разобрался со всеми особенностями программы и выполняет действия, не предусмотренные разработчиком. С приобретением опыта таких неприятных сюрпризов становится меньше.

Интерфейс программы локализован только частично; чтобы не возникало путаницы, все названия пунктов меню приводятся без изменений.

Включаем Multitrack

Режим интерфейса по умолчанию устанавливается в Startup Mode (Инструменты > Параметры > GUI) или если он нужен временно на текущий проект, то выбираем Правка > MultiTrack режим (Ctrl+M). Хотя есть еще один способ. Если во время воспроизведения клипа активировать запись фрагмента (пуск/остановка

клавишей R, активируется счетчик кадров), то по окончании записи появится меню, где можно выбрать вариант просмотра/редактирования в Multitrack [View/edit events in multitrack window (test)]. Режим, как видим, тестовый, но он работает.

При переходе в Multitrack появится окно с настройками Multitrack details, в котором необходимо установить параметры проекта: видео (разрешение и частота кадров), аудио (частота, каналы и т.п.), результат (декодер, формат файла, аудиоформат). В принципе, сейчас важны первые два: они могут влиять на дальнейшую работу с некоторыми эффектами. Чтобы установки использовались по умолчанию, достаточно поставить флажок Always use these values [Всегда использовать эти значения]. Впоследствии что-то можно переопределить (для текущего проекта — в Инструменты > Change width, height and audio values... [Изменить ширину, высоту и параметры аудио] или глобально в Инструменты > Параметры > Multitrack/Render). Два флажка активированы по умолчанию и требуют отдельного объяснения. Так, Enable backing audio track [Активировать фоновую дорожку аудио] добавляет в проект еще одну общую аудиодорожку, которую можно использовать для фоновой музыки. Флажок Audio track per video track включает аудиодорожку в отдельном видео.

По умолчанию проект создает резервную копию каждые 120 секунд; на относительно медленных системах это тормозит работу. Установки можно изменить в Multitrack/Render, приказав делать бэкап в другое время, отключить совсем или, наоборот, заставить делать после каждого изменения [After every change].

Внешний вид в режиме MultiTrack при первом знакомстве выглядит пугающе, но все логично — нужно просто разобраться, потратив на это некоторое время, а если вы уже работали с другими редакторами, то и изменить некоторые привычки. Затем все пойдет как по маслу. Поэтому вначале разберемся с интерфейсом.

Интерфейс MultiTrack

Функционально, кроме меню и панели управления можно выделить несколько областей. Первой идет так называемая manipulation area [область манипуляции], состоящая из трех окон: предварительного просмотра, polytopgrh и сведений. В первом показывается сам клип или результат его обработки, в последнем — советы. Подсказки помогают освоиться быстрее, показывая информацию по текущему действию, но полезны они будут только при знании базового английского (не локализованы); их можно убрать при помощи кнопки Expanded/Compact View или горячей клавиши d.

Часть, находящаяся посередине, состоит из нескольких вкладок и переключает свой вид автоматически в зависимости от текущей операции. Именно она при первом знакомстве с *LiVES* вносит изрядную путаницу с назначением, поэтому некоторое время придется осваиваться, а не локализованная документация здесь не слишком помогает. Вкладки можно выбирать самостоятельно, но эта операция не имеет смысла: в некоторых мы увидим

НА МНОГО ДОРОЖЕК

пустое окно без каких-либо настроек, и чтобы они там появились, нужно выполнить некоторые действия.

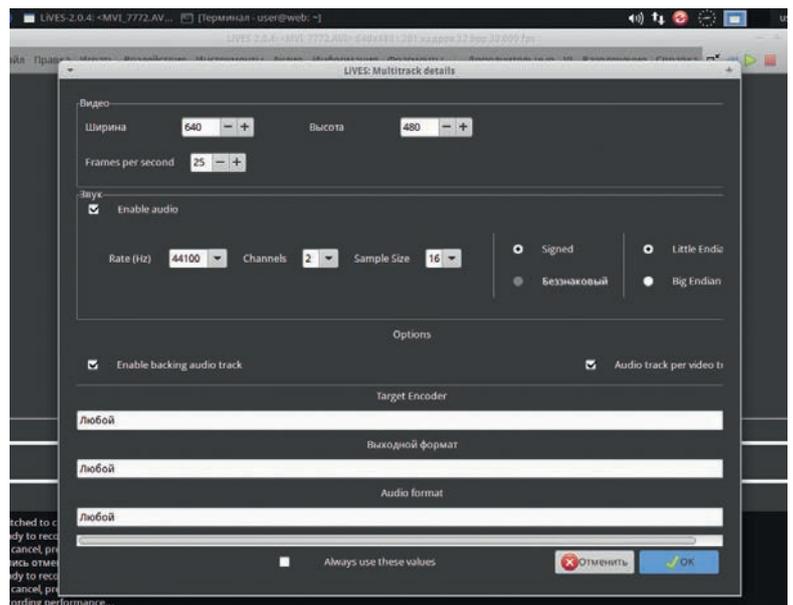
После загрузки активна первая вкладка Clip, где мы увидим все ролики проекта. Если дважды щелкнуть по клипу, то попадаем обратно в Clip Editor (как вариант, выбрать Edit/encode in clip editor в контекстном меню). Из Clip можно удалить видео из списков проекта и просмотреть информацию. Вкладка In/Out позволяет задать первый и последний кадр клипа, который будет использоваться при дальнейшей обработке. Если сюда зайти, внутри пусто и назначение непонятно. Но есть несколько способов: выбрать на видео в Clip пункт контекстного меню Adjust start and end points или на видео, помещенном на временную шкалу, дважды щелкнуть (оно сразу помечается крестиком), или выбрать в контекстном меню пункт Выберите этот блок. После любого действия вкладка In/Out открывается автоматически. Далее просто отмечаем нужный кадр и при необходимости фиксируем его установкой флажка Anchor start/end. Если видео уже помещено на временную шкалу, мы увидим, что оно уменьшается в размере, а значит, все треки проекта придется подгонять повторно.

В остальных вкладках — FX stack, Эффекты, Переходы состояний, Compositors и Params — показываются специфические настройки эффектов, если они применялись к выбранному видео, об этом поговорим далее.

Чуть ниже располагается временная шкала, в которой, собственно, и редактируются аудио и видео. В самом верху — горизонтальная линейка с временными отметками [timeline]. Если щелкнуть по ней в любом месте один раз, установится метка, и воспроизведение будет начинаться с этой точки. Когда проект большой, нередко требуется обозначать ключевые места, чтобы к ним была возможность быстро вернуться. Для этого во время воспроизведения необходимо нажимать клавишу **m**, и на шкале появится синяя метка; таких меток можно ставить сколько угодно. Чтобы выделить участок, также следует щелкнуть по timeline и провести по линии до нужного места, или просто указать время начала и окончания в поле Выбор.

Если вставлено большое количество объектов и все они не умещаются, появляется полоса прокрутки. При следующем открытии программы файлы будут автоматически масштабированы под ширину окна LiVES. При помощи клавиш **Ctrl +/-** можно растянуть или уменьшить timeline, как это делается в большинстве редакторов.

И в самом низу окна располагается область уведомлений, в которой можно наблюдать за ходом текущей операции — если она мешает, ее можно уменьшить или совсем отключить.



Работаем с дорожками

По умолчанию на временной шкале располагается дорожка фоновой музыки (Backing audio) и две видеодорожки (Layer). Общую дорожку фонового звука, если она не нужна, при текущей обработке можно скрыть, чтобы она не занимала место (Вид > Show backing audio track). Чтобы добавить новый слой видео, следует

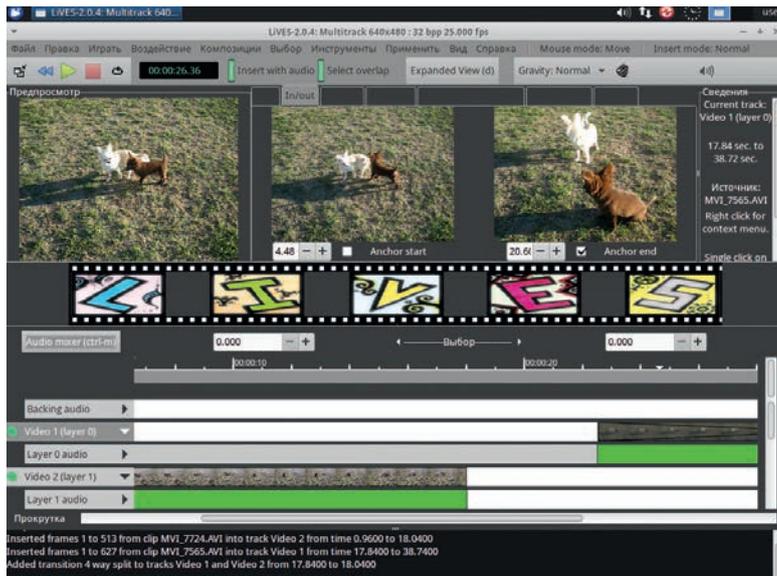
обратиться к меню Композиции > Add Video Track at Rear/Add Video Track at Front (**Ctrl+T**/**Ctrl+Shift+T**). Для удобства в больших проектах дорожки лучше переименовать (Композиции > Rename current track),

чтобы было сразу понятно их назначение. Кроме того, фоновую музыку может иметь каждая дорожка; для этого необходимо ее отметить и выбрать Композиции > Make Backing Audio Current Track (**Ctrl+B**).

Чтобы добавить видео на дорожку, достаточно в окне Clip захватить его мышью и перенести в нужное место; другой вариант — выделить видео щелчком мыши, а затем на дорожке использовать пункт контекстного меню Вставить здесь или выделить видео и нажать клавишу **I**. В последнем случае файл будет добавлен на первую дорожку в начало — это очень удобно, если нужно просто их расставить по порядку, меньше подгоняя вручную. И, главное, места для видео в точке вставки (если она располагается между двумя клипами) должно хватать. Если клип

» Настройки проекта.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



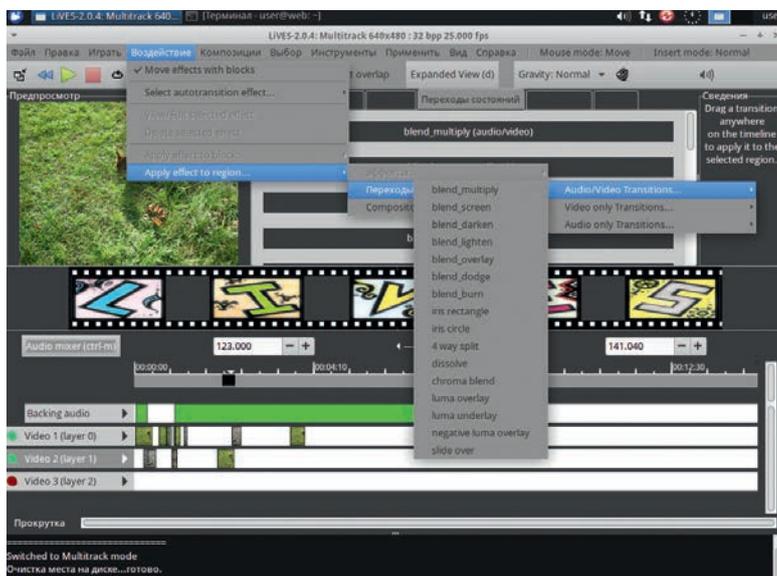
➤ **Задаем началь- ный и конечный кадры видео.**

будет больше, то он будет обрезан. Поэтому лучше посмотреть доступное время по timeline, и если его не хватает, сдвинуть остальные видео, освободив достаточно места. Это сделать очень просто. Выделяем нужную область на timeline при помощи мыши или счетчика Выбор и вставляем в это место разрыв при помощи Insert gap in selected track/time или Insert gap in current track/ selected time.

Как и при использовании Clip Editor, в проект можно вставить титры, пустой или цветной кадр, сгенерировать видео из изображений. Все эти действия доступны в меню Инструменты > Создать и описаны в предыдущей части учебника (LXF175).

Обратите внимания на два переключателя в меню: Insert with audio [Вставка видео вместе с аудио] и Select overlap (позволит выбрать на timeline только место, где перекрываются треки, это полезно при создании эффектов перехода и композитинге). Нажав на стрелку в каждом слое, можно увидеть аудиодорожку.

➤ **Выбираем эффект перехода.**



«В этой работе помогают горячие клавиши — не лишне их изучить.»

В этой работе сильно помогают горячие клавиши — не лишне будет их изучить и использовать. Самые употребительные показаны во врезке на стр. 91.

Каждый трек может иметь свое аудио (если выбран Insert with audio), плюс фоновая музыка; чтобы управиться со всем этим, LIVES предлагает простой микшер, вызываемый кнопкой Audio mixer (Ctrl+M). С его помощью можно выставить уровень громкости для каждого источника. Обратите внимание на переключатель Gang layer audio: если он установлен (горит зеленым), то для всех слоев устанавливается одно значение громкости. Поэтому если необходимо для разных каналов установить свой уровень, то деактивируем его. Еще один флажок, Invert backing audio and layer volumes, позволяет инвертировать значения музыки в видео и фоновой.

Чем меньше номер слоя видео, тем выше в кадре он будет. То есть если клипы перекрывают друг друга, мы будем видеть только то, что находится в Layer 0, если, конечно, не применены эффекты перехода или compositor. На каждой дорожке имеются переключатели (красный/зеленый кружок), позволяющий включать текущий трек в область [region]; далее они нам понадобятся.

Возможное действие на клипе, помещенном на временной шкале, изменяется в зависимости от установленного значения Mouse move в Select или Move. В первом случае мы можем выделить любой участок видео, во втором — видео захватывается, и его можно перенести в любое место и расставить как нужно.

Обрабатываем видео

Клип, размещенный на временной шкале, в терминологии LIVES называется блоком — это наименьшая единица, с которой может работать наш редактор. При этом блок может быть и один кадр, если его отобразить при помощи In/Out или обрезать, используя пункт Split block here контекстного меню (вариант — установить курсор и нажать комбинацию Ctrl+S). Разрезать все видео на всех треках в точке курсора можно при помощи Композиции > Split selected video track. Лишняя часть затем обрабатывается отдельно, удаляется или перемещается.

Чтобы произвести какую-либо операцию над блоком, достаточно дважды по нему щелкнуть, после чего автоматически активируется вкладка In/Out.

Эффекты реального времени, предлагаемые в режиме Clip Editor, сейчас нам недоступны, только обычные [rendered]. Применить эффект очень просто. Выделяем блок и переходим в Воздействие > Apply effect to block, где отмечаем нужный эффект. После этого активируется вкладка FX effect, в которой мы можем просмотреть список всех эффектов блока (переход по Prev filter map/Next filter map) и удалить.

Если эффектов несколько, то их можно перемещать, меняя приоритет при помощи Insert before/Insert after. Для настройки эффекта (если это предусмотрено) необходимо его отметить в FX effect и перейти в Params. В окне предварительного просмотра можно будет сразу оценить результат. Причем если в Params доступна полоса Время (например, для colour correction), то для разных частей одного блока можно использовать разные установки эффекта. Просто в разных точках временной шкалы указываем свои значения и нажимаем кнопку Применить. Впоследствии между точками, в которых использованы разные параметры, можно просто перемещаться при помощи кнопок Prev node/Next node; чтобы удалить точку, используем Delete node.

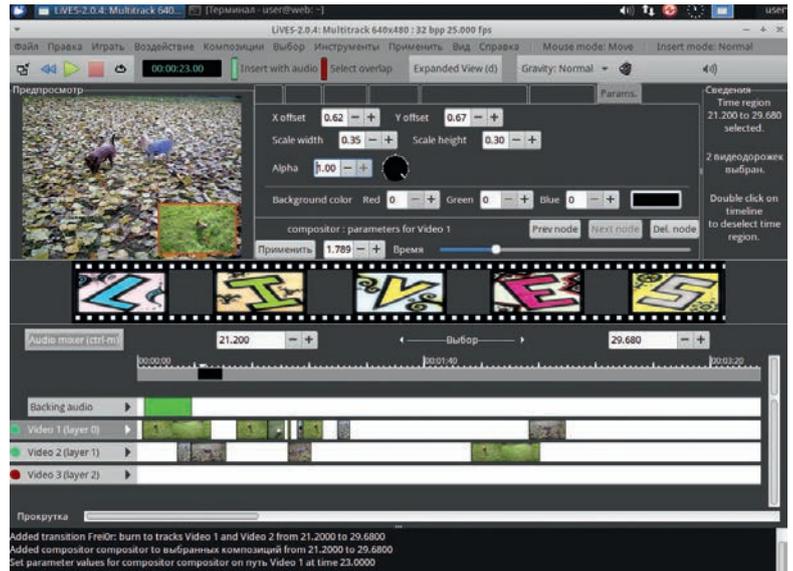
➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Есть здесь еще интересный момент. По умолчанию при перемещении блока эффекты также остаются вместе с ним, что в общем логично, но не всегда. Например, в разных частях проекта используются разные эффекты, а поэтому, перенеся блок с другого места, мы можем испортить общую картину. Но это решается очень просто: достаточно снять флажок Воздействие > Move effect with block, и блок будет перенесен в новое место в «чистом» виде.

Создание переходов

Склеивая несколько кусков видео, необходимо создать целостное впечатление: без переходов резкие скачки сюжета смотрятся не очень красиво. Правда, увлекаться ими тоже не стоит: это сильно мешает восприятию и даже считается дурным тоном. Чтобы создать переход, необходимо видео разместить в разных слоях, при этом они должны немного пересекаться. Затем требуется создать область. В терминах LiVES область представляет собой сочетание нескольких треков и выделенной части на timeline. Треки включаются в регион при помощи переключателя (зеленый кружок, Ctrl+Space) или, если все, Выбор > Select all video track. Далее выделяем на временной шкале участок между блоками, для которых создается переход; становится активным пункт меню Воздействие > Apply effect to region. Перейдя сюда, обнаружим два подпункта. В Переходы состояний собственно и выбирается нужный эффект для аудио и видео (отдельно или одним пунктом), после чего активируется вкладка Params, и можно настраивать параметры.

В Compositor только один подпункт; после его активации попадаем в одноименную вкладку в rotomorph. Здесь вручную настраивается взаимодействие видео на треках, в том числе и переход между ними; причем, в отличие от эффектов перехода, Compositor может охватывать более двух треков. Для определенного времени указываются установки относительного сдвига по оси X и Y, размер кадра видео, располагаемого в слое выше, цвет рамки, прозрачность и т.п. Участок задается при помощи



Режим Compositor.

переключателей или просто выделением мышью в нужном месте квадрата в окне предпросмотра.

Процесс работы аналогичен применению эффектов. Выбираем определенное время, настраиваем и нажимаем Применить. Затем операцию повторяем для следующего временного отрезка. При помощи Compositor можно также создавать такой эффект, как «кадр в кадре».

Расположить видео по нескольким дорожкам с большой точностью очень сложно: кроме перекрытий, могут появиться и пробелы. Но в режиме региона их легко устранить автоматически, достаточно выбрать пункт Ctrl+G, Ctrl+F.

Конечно, некоторое время придется осваиваться со особенностями мультиторочечного режима LiVES, но впоследствии можно будет создавать отличные проекты. LXF

Комбинации горячих клавиш при работе с LiVES

Клавиша	Значение	Клавиша	Значение
P	Воспроизведение	Ctrl+PageUp и Ctrl+PageDown	Выбор видео в окне Clips
Y	Воспроизведение только выбранного	Ctrl+Left/Right	Перемещение курсора влево/право на временной шкале
S	Воспроизведение в отдельном окне	Ctrl+Up/Down	Изменение дорожки
F	Переключение в полный экран	Ctrl+I	Вставка выбранного клипа в текущую позицию на временной дорожке.
M	Метка на временной шкале	Ctrl+A	Выделить всю временную шкалу
W	Сброс позиции курсора	Ctrl+B	Добавление фоновой музыки к дорожке
I	Вставка видео на трек	Ctrl+S	Обрезка трека
R	Запись фрагмента	Ctrl+Space	Включение трека в область
D	Переключение Expanded/Compact View	Ctrl+T/Ctrl+Shift+T	Добавление видеотреков в слой выше или ниже
Ctrl+M	Переход из Clip Editor в режим MultiTrack, в режиме MultiTrack — вызов аудиомикшера	Ctrl+R	Рендеринг проекта
Ctrl+S	Разделение блока под курсором	Ctrl+G	Заполнить пустое место
Ctrl+ +/-	Увеличение или уменьшение масштаба временной шкалы	Ctrl+F	Заполнить первое пустое место в области



LMD: Защищаем

Игорь Штомпель обнаружил надежный щит, заслоняющий нашу любимую ОС от посягательств злоумышленников.



Наш эксперт

Игорь Штомпель каждый день открывает в Linux что-то новое и использует его где только возможно. А Unity ему в этом совсем не мешает.

Linux известен своей надежностью и высоким уровнем безопасности как в качестве рабочей станции, так и сервера. Но вот web-пространство полно кибер-угроз...

LMD на страже системы

Одним из решений, помогающих сделать контакты с web-пространством безопаснее, является *Linux Malware Detect*, или *LMD*. Данное программное обеспечение представляет собой сканер, который призван обнаруживать вредоносное программное обеспечение на статических страницах web-сайтов и в файлах web-приложений. В отличие от «традиционных» вирусов, которые «заражают» и «размножаются», это ссылки на вредоносные скрипты, указывающие на другие сайты (например, при XSS-атаках), или код, внедренный злоумышленником на защищаемом сайте (например, PHP-инъекция).

Деятельность *LMD* основана на довольно внушительном перечне возможностей, которые заявлены разработчиками. И, соответственно, для более полного понимания работы *LMD* необходимо немного подробнее рассмотреть эти возможности, в том числе методы борьбы с вредоносным программным обеспечением.

Базой, которая позволяет *LMD* обнаруживать вредоносное ПО, служат MD5- и HEX-сигнатуры. В частности, для выявления файлов, имеющих вредоносное содержимое и прошедших обфускацию, применяется статистический анализ исследуемых файлов. Обфускация — это изменение исходного кода или исполняемого файла, при сохранении их функциональности, с целью усложнения их анализа и понимания. Более подробную информацию об обфускации можно получить, например, в Википедии [1]. Соответственно, *LMD* в состоянии удалять из сканируемых файлов вирусные инъекции.

LMD обладает такими «стандартными» возможностями для антивирусного ПО, как обновление сигнатур, отправка подозрительных файлов на сервер *LMD*, ведение журнала и т.п. А помимо того, *LMD* обладает рядом специфических особенностей: например, он в состоянии определять наличие в системе *ClamAV* и использовать его в качестве сканера. Или осуществлять сканирование измененных за определенное количество дней (определяемое пользователем) файлов в заданном каталоге в ручном режиме.

Еще одной особенностью *LMD* является процесс формирования базы вредоносного ПО. Как заявляют сами разработчики на своем официальном сайте [2], их антивирусный сканер позволяет не просто определять вирусы (здесь и далее, будем употреблять слово «вирус» как синоним вредоносного ПО) на базе хэшей и сигнатур, но и защищать от основных web-угроз, представляющих опасность в настоящее время. Таким образом, проект *LMD* концентрирует основное внимание на угрозах для серверных решений.

Соответственно, это наложило отпечаток на процесс пополнения антивирусной базы. Разработчики *LMD* организовали такое пополнение, выделив четыре источника сигнатур.

Первый — это данные из сети Network Edge IPS (здесь и далее — название источников, данное разработчиками). Дело

в том, что разработчик *LMD* управляет хостингом, на котором размещено свыше 35 000 сайтов, что позволяет ему пополнять коллекцию вредоносного ПО, выявляемого на основе работы IPS. Второй — данные, предоставленные сообществом (Community Data). Прежде всего имеются в виду порталы сообществ, посвященных вредоносному ПО — например, такие, как CLEAN MX [3] и Malware Domain List [4]. Информация с последних собирается, исследуется, выявленное вредоносное ПО классифицируется и для него создаются соответствующие сигнатуры. Третий — данные проекта *ClamAV* (ClamAV). Разработчики *LMD* осуществляют мониторинг сигнатур проекта *ClamAV* (MD5 и HEX) на предмет подходящих для необходимого функционирования *LMD* (напомним, что сказали выше: основное внимание *LMD* уделяет защите серверных решений), которые по мере обнаружения добавляют в свои базы сигнатур. Кстати, сами разработчики приводят следующие данные на своем официальном сайте: ~400 сигнатур получено от проекта *ClamAV* и более 1100 сигнатур было отправлено проекту *ClamAV*. Четвертый — пользователи (User Submission). Выше уже говорилось о том, что *LMD* предоставляет пользователям возможность отправки подозрительных файлов на серверы *LMD*. Именно это и является одним из источников пополнения сигнатур проекта. По данным разработчиков, пользователи делают 30–50 отправок каждую неделю.

В завершение рассмотрения возможностей *LMD* отметим, что база сигнатур проекта, как проиллюстрировано на рис. 1, содержит 11 304 записей. Среди них, 9432 — это MD5-сигнатуры, а 1872 — HEX-сигнатуры. Файлы баз данных можно найти в каталоге `/usr/local/maldetect/sigs: hex.dat, md5.dat, rfxn.hdb, rfxn.ndb`. Кстати, следить за обновлением сигнатур *LMD* можно с помощью специального RSS-канала проекта [5].

Установка LMD

Перейдем к рассмотрению установки *LMD*. Чтобы раздобыть *Linux Malware Detect*, его необходимо загрузить с официального сайта. Ведь данное программное обеспечение отсутствует в репозиториях многих популярных дистрибутивов. В репозиториях Linux Mint 15, который мы будем использовать для установки и работы с *LMD*, его также не имеется.

Итак, перейдите в каталог, в котором вы хотите произвести процесс установки. Например, остановим свой выбор на каталоге `~/Install`. Теперь загрузим *LMD* с официального сайта (общий объем архива ~792 КБ):

```
wget http://www.rfxn.com/downloads/maldetect-current.tar.gz
```

Далее, распакуем полученный архив:

```
tar -xzf maldetect-current.tar.gz
```

Программа будет распакована в каталог `maldetect-1.4.2`. Перейдем в него и запустим установку:

```
sudo ./install.sh
```

Как видно на рис. 1, *LMD* во время установки, среди прочего, обновляет сигнатуры. И обратите внимание, что версия программы в выводе показана как

```
Linux Malware Detect v.1.4.1
```

Linux от угроз

хотя, как показано выше, мы загружали версию 1.4.2, в которую «превратится» наш экземпляр данного ПО после обновления. Что можно будет видеть в выводе *LMD* в дальнейшем.

Кстати, на рис. 1 также видно, что программа устанавливается в каталог `/usr/local/maldetect`, а также то, что создается файл `/etc/cron.daily/maldet`. Последний представляет собой вариант ежедневных заданий для *Cron*.

Познакомившись внимательнее с содержимым ежедневных заданий для *Cron*, которые автоматически создал *LMD* на этапе установки, мы обнаружим, что программа будет каждый день проверять выход новых версий и сигнатур. В этом можно убедиться, посмотрев на рис. 2.

Кроме того, пролистав файл с заданиями для *Cron* далее, мы увидим, что в качестве целей для сканирования указаны стандартные каталоги для популярного серверного программного обеспечения (в порядке перечисления в файле): `ensim`, `psa`, `DirectAdmin`, `cpanel`, `interworx`, `Apache` (`/var/www/html/`; `/usr/local/apache/htdocs`).

LMD в действии!

LMD — это консольный инструмент, у которого один-единственный интерфейс: CLI, командная строка. Рассмотрим ряд приемов, которые позволяют его эффективно использовать.

Обратите внимание на то, что при попытке провести «публичное сканирование» — в терминах *LMD* это означает «без прав суперпользователя» — программа сообщит о том, что такой вид сканирования отключен:

```
public scanning is currently disabled, please contact your system
administrator to enable public scan in conf.maldet.
```

При желании, включить публичное сканирование можно путем редактирования конфигурационного файла *LMD*, который имеет следующее местоположение: `/usr/local/maldetect/conf.maldet`. Для этого достаточно изменить значение опции `public_scan` на 1 (по умолчанию оно равно 0). Но делать этого не рекомендуется, так как *Linux Malware Detect* не сможет сканировать каталоги, доступ к которым есть только у суперпользователя, что значительно снижает эффективность работы программы и дает

```
Linux Malware Detect v1.4.1
(C) 2002-2013, R-fx Networks <proj@r-fx.org>
(C) 2013, Ryan MacDonald <ryan@r-fx.org>
inotifywait (C) 2007, Rohan McGovern <rohan@mcgovern.id.au>
This program may be freely redistributed under the terms of the GNU GPL

installation completed to /usr/local/maldetect
config file: /usr/local/maldetect/conf.maldet
exec file: /usr/local/maldetect/maldet
exec link: /usr/local/sbin/maldet
exec link: /usr/local/sbin/lmd
cron.daily: /etc/cron.daily/maldet

maldet(2746): {sigup} performing signature update check...
maldet(2746): {sigup} local signature set is version 201205035915
maldet(2746): {sigup} new signature set (201310259491) available
maldet(2746): {sigup} downloaded http://www.rfxn.com/downloads/md5.dat
maldet(2746): {sigup} downloaded http://www.rfxn.com/downloads/hex.dat
maldet(2746): {sigup} downloaded http://www.rfxn.com/downloads/rfxn.ndb
maldet(2746): {sigup} downloaded http://www.rfxn.com/downloads/rfxn.hdb
maldet(2746): {sigup} downloaded http://www.rfxn.com/downloads/maldet.cle
maldet(2746): {sigup} signature set update completed
maldet(2746): {sigup} 11304 signatures (9432 MD5 / 1872 HEX)
```

» Рис. 1. *LMD* успешно обновил сигнатуры в процессе установки.

```
/etc/cron.daily/maldet 804/1943
#!/bin/bash

# clear quarantine/session/tmp data every 14 days
/usr/sbin/tmpwatch 336 /usr/local/maldetect/tmp >> /dev/null 2>&1
/usr/sbin/tmpwatch 336 /usr/local/maldetect/sess >> /dev/null 2>&1
/usr/sbin/tmpwatch 336 /usr/local/maldetect/quarantine >> /dev/null 2>&1
/usr/sbin/tmpwatch 336 /usr/local/maldetect/pub/*/* >> /dev/null 2>&1

# check for new release version
/usr/local/maldetect/maldet -d >> /dev/null 2>&1

# check for new definition set
/usr/local/maldetect/maldet -u >> /dev/null 2>&1

# if were running inotify monitoring, send daily hit summary
if [ "$(ps -A --user root -o "comm" | grep inotifywait)" ]; then
    /usr/local/maldetect/maldet --alert-daily >> /dev/null 2>&1
else
    # scan the last 2 days of file changes
    if [ -d "/home/virtual" ] && [ -d "/usr/lib/openscenter" ]; then
```

неверное представление об уровне безопасности исследуемого содержимого.

Сканирование определенного каталога, например, каталога `scan` в текущем, выполняется следующим образом:

```
$sudo maldet -a scan
```

Вывод работы этой команды представлен на рис. 3. Нам, в данном случае, интересны последние две строки вывода. Первая — `maldet(9717): {scan} scan completed on scan: files 111, malware hits 0, cleaned hits 0`

говорит о том, что просканировано 111 файлов, в которых вирусов не обнаружено [`malware hits 0`], и, соответственно, не было вылено ни одного файла [`cleaned hits 0`].

Вторая строка —

```
maldet(9717): {scan} scan report saved, to view run: maldet
--report 110213-1526.9717
```

сообщает, что отчет о сканировании сохранен [`scan report saved`] и его можно просмотреть с помощью соответствующего параметра команды `maldet` (`--report`). Отчеты сохраняются в каталоге `/usr/local/maldetect/sess`.

Что ж, примем предложение *LMD* и посмотрим отчет:

```
$sudo maldet --report 110213-1526.9717
```

или просто используем опцию `-e` (актуально в данном виде только для последнего сканирования):

```
$sudo maldet -e
```

Как показано на рис. 4, в отчете мы увидим идентификатор сканирования (`SCAN ID`), дату и время проведения (`TIME`), сканируемый каталог (`PATH`), общее количество сканируемых файлов (`TOTAL FILES`), количество обнаруженных вирусов (`TOTAL HITS`), количество вычлеченных файлов (`TOTAL CLEANDED`).

Кроме того, *LMD* способен, как мы отметили выше, выявлять файлы, измененные в течение указанного количества дней. Для примера, сканируем файлы в каталоге `scan`, в которые вносились изменения за последние три дня:

```
$sudo maldet -r scan 3
```

где `-r` — соответствующая опция *LMD*, 3 — число дней.

» Рис. 2. Фрагмент ежедневных заданий *LMD* для *Cron*.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

```
gor@point ~ $ sudo maldet -a scan
Linux Malware Detect v1.4.2
  (C) 2002-2013, R-fx Networks <proj@r-fx.org>
  (C) 2013, Ryan MacDonald <ryan@r-fx.org>
  notifywait (C) 2007, Rohan McGovern <rohan@mcgovern.id.au>
  this program may be freely redistributed under the terms of the GNU GPL v2

aldet(9717): {scan} signatures loaded: 11304 (9432 MD5 / 1872 HEX)
aldet(9717): {scan} building file list for scan, this might take awhile...
aldet(9717): {scan} file list completed, found 111 files...
aldet(9717): {scan} 111/111 files scanned: 0 hits 0 cleaned
aldet(9717): {scan} scan completed on scan: files 111, malware hits 0, cleaned hits 0
aldet(9717): {scan} scan report saved, to view run: maldet --report 110213-1526.9717
```

➤ Рис. 3. Вывод LMD в процессе сканирования.

Сканирование в фоновом режиме

Этой возможностью будет кстати при проведении сканирования большого количества файлов. Чтобы его запустить, необходимо воспользоваться опцией **-b**:

```
$sudo maldet -b -a scan
```

Следить за процессом сканирования можно, например, с помощью опции **-l**, которая позволяет просматривать логи *Linux Malware Detect*:

```
$sudo maldet -l
```

Обратите внимание, что о завершении сканирования можно говорить при появлении записи о сохранении отчета (пример последней приведен выше):

```
...scan report saved, to view run:...
```

в логах *LMD*.

Действия при обнаружении вирусов

Здесь возможны несколько вариантов действий: при выключенном или включенном карантине. Рассмотрим первый случай.

Допустим, после проведенного сканирования —

```
$sudo maldet -a scan
```

— *LMD* обнаружил вирус. Как, например, демонстрирует рис. 5: в выводе команды **maldet** мы видим, что обнаружен один зараженный файл.

Просмотрим отчет о результате текущего сканирования, которое привело к обнаружению вредоносного ПО:

```
$ sudo maldet --report 110313-1531.30360
```

Мы видим, что был обнаружен вирус в файле **serv.php**, HEX-сигнатура которого соответствует записи **gzbases64.inject.unclassified.18** в базах *LMD*.

Пришла пора задействовать возможности карантина *LMD* по обезвреживанию вируса. «Натравим» на последний *LMD* с помощью опции **-q**:

```
$sudo maldet -q 110313-1531.30360
```

Как показано на рис. 6, борьба с обнаруженным вирусом завершилась успешно. Но заглянем «под капот» и посмотрим, что получилось по факту.

В каталоге **scan**, в котором и был обнаружен зараженный файл, ничего не изменилось. Файл **serv.php** остался на месте. А вот в карантине *LMD* (**/usr/local/maldetect/quarantine**) появился файл с именем **serv.php.28885.info**.

Данный файл имеет следующее содержание:

```
GNU nano 2.2.6 Файл: /usr/local/maldetect/scan/scan
Linux Malware Detect scan report for point:
SCAN ID: 110213-1526.9717
TIME: нояб. 2 15:26:17 +0400
PATH: scan
TOTAL FILES: 111
TOTAL HITS: 0
TOTAL CLEANED: 0

=====
Linux Malware Detect v1.4.2 < proj@rfxn.com >
```

➤ Рис. 4. Просмотр отчета LMD.

```
igor igor 600 scan/serv.php
```

Нетрудно догадаться, что это имя пользователя и его группа, права доступа к файлу, полный путь к файлу **serv.php**.

Таким образом, запись о вредоносном файле помещена в карантин, а сам файл остался там, где и был. Но его можно удалить вручную. Таким образом, при выключенном карантине *LMD* позволяет обрабатывать найденные файлы с вредоносным содержанием самостоятельно.

Кроме того, как было видно на рисунках, если карантин не включен, то как в процессе сканирования и обнаружения вируса, так и при просмотре соответствующего отчета *Linux Malware Detect* «просигнализирует» о том, что поддержка карантина отключена:

```
quarantine is disabled!
```

Для автоматической нейтрализации вредоносного ПО карантин необходимо включить. Важно еще и то, что при отключенном карантине зараженный файл не сможет быть удален в ситуации, когда лечение файла (например, чистка от вирусного содержания) невозможно. В этом можно убедиться, просмотрев конфигурационный файл *LMD*. В нем, по умолчанию, параметр **quar_clean**, который позволяет осуществлять лечение файлов, равен 1. При этом в карантине *LMD* (**/usr/local/maldetect/quarantine**) будет лишь появляться файл с информацией о зараженном файле.

Итак, чтобы включить карантин, надо отредактировать конфигурационный файл *Linux Malware Detect*. Для этого даем следующую команду:

```
$sudo mc /usr/local/maldet
```

Далее находим файл **conf.maldet** и вносим следующие изменения:

```
quar_hits=1
```

Сохраняем изменения и закрываем конфигурационный файл *LMD*.

Теперь проведем сканирование со включенным карантинном:

```
$sudo maldet -a scan
```

В этой ситуации, в выводе данной команды мы увидим информацию об обнаружении двух вирусов. Чтобы прояснить ситуацию, посмотрим отчет:

```
$sudo --report 110313-1546.646
```

В отчете мы увидим, что *LMD* обнаружил два файла с вредоносным содержанием, которые находятся в карантине. Что ж, обратимся к содержимому «карантинного» каталога *LMD*:

```
ls /usr/local/maldetect/quarantine
```

Вывод этой команды показывает, что в карантине *LMD* — три файла:

```
serv.php.29046.info
```

```
serv.php.7449
```

```
serv.php.7449.info
```

Изучив их содержимое (например, с помощью встроенного редактора *Midnight Commander* — клавиша F3) и обратив внимание на объем файлов, мы увидим, что второй файл является копией зараженного файла, а два остальных содержат информацию, о которой речь шла уже выше:

```
igor igor 600 scan/serv.php
```

При этом сам файл **serv.php** с вредоносным содержанием удален из каталога **scan**.

Итак, при включенном карантине зараженные файлы, лечение которых невозможно, удаляются из сканируемого каталога в карантин *LMD*. А при выключенном карантине — нет.

Отправка подозрительного файла

Как мы упоминали выше, одним из способов пополнения антивирусных баз *LMD* является деятельность пользователей. Им

в помощь разработчики предусмотрели возможность отправки подозрительных файлов для исследования на свой сервер (rfxn.com). После чего файлы подвергнутся изучению, и при выявлении вредоносного содержимого соответствующие записи будут добавлены в антивирусные базы.

Как же отправить подозрительный файл? Для этого достаточно воспользоваться опцией `-c` и скомандовать:

```
$sudo maldet -c /home/каталог_текущий_пользователь/серв.
php
```

где `/home/каталог_текущий_пользователь` — домашний каталог текущего пользователя, `серв.php` — подозрительный файл.

Обновление Linux Malware Detect

Выше мы уже говорили о том, что во время установки *LMD* создает соответствующие файлы для автоматического обновления с помощью *Cron*. И, кстати, в действенности этого можно убедиться, если посмотреть на рис. 1, на котором общее количество сигнатур в базах *LMD* равно 11 304, и на рис. 5, на котором общее количество сигнатур уже увеличилось до 11 337. Вот так, незаметно, и произошло обновление.

Для проверки выхода новой версии *LMD* в «ручном» режиме предусмотрена опция `-d`:

```
$sudo maldet -d
```

Если будет обнаружена новая версия, вы сможете обновить *LMD*, а если нет, то увидите в выводе, среди прочего, сообщение о том, что у вас установлена последняя версия:

```
latest version already installed.
```

Обновить антивирусные базы *LMD* можно при посредстве опции `-u`:

```
$sudo maldet -u
```

При этом, если базы будут в актуальном состоянии, вы увидите в выводе, среди прочего, следующее сообщение:

```
latest signature set already installed
```

И, конечно, при обнаружении новых сигнатур на сервере своих разработчиков *LMD* произведет обновление автоматически.

Мониторинг изменений

С выходом ядра Linux 2.6.13 стала доступна новая подсистема, которая позволяет осуществлять мониторинг файловой системы (открытие, перемещение и т. д.) в реальном времени. Она получила название *Inotify*.

Разработчики *Linux Malware Detect* сделали доступной возможность взаимодействия своей программы с *Inotify*. А раз так, то и воспользуемся ею.

Для начала проверим, установлен ли пакет *inotify-tools* (по умолчанию, в Linux Mint этот пакет не установлен):

```
$sudo dpkg --status inotify-tools
```

В нашем случае, пакета не оказалось. Установим его:

```
$sudo apt-get install inotify-tools
```

Первая попытка запуска мониторинга *LMD* с помощью *Inotify*:

```
$sudo maldet --monitor ~/scan
```

оказалась неудачной. Причина оказалась в том, что *LMD* поставляется совместно с двоичными файлами *inotifywait* и *libinotifytools.so.0*, но работает с ними «из коробки» некорректно. Для исправления ситуации можно воспользоваться оригинальным решением проблемы, приведенном на одном из форумов (<http://www.howtoforge.com/forums/showthread.php?t=58440>). В частности, необходимо внести следующие изменения в конфигурационный файл `/usr/local/maldetect/internals.conf`: строку

```
inotify=$inspath/inotify/inotifywait
```

заменить на

```
inotify=/usr/bin/inotifywait
```

Тогда последней мы будем использовать, вместо поставляемых с *LMD* двоичных файлов *Inotify*, ту версию, которую установили из репозитория.

```
igor@point ~ $ sudo maldet -a scan
Linux Malware Detect v1.4.2
(C) 2002-2013, R-fx Networks <proj@r-fx.org>
(C) 2013, Ryan MacDonald <ryan@r-fx.org>
inotifywait (C) 2007, Rohan McGovern <rohan@mcgovern.id.au>
This program may be freely redistributed under the terms of the GNU GPL v2

maldet(30360): {scan} signatures loaded: 11337 (9465 MD5 / 1872 HEX)
maldet(30360): {scan} building file list for scan, this might take awhile...
maldet(30360): {scan} file list completed, found 112 files...
maldet(30360): {scan} 112/112 files scanned: 1 hits 0 cleaned
maldet(30360): {scan} scan completed on scan: files 112, malware hits 1, cleaned hits 0
maldet(30360): {scan} scan report saved, to view run: maldet --report 110313-1531.30360
maldet(30360): {scan} quarantine is disabled! set quar_hits=1 in conf.maldet or to quarantine
ults run: maldet -q 110313-1531.30360
```

► Рис. 5. *LMD* обнаружил зараженный файл.

```
igor@point ~ $ sudo maldet -q 110313-1531.30360
Linux Malware Detect v1.4.2
(C) 2002-2013, R-fx Networks <proj@r-fx.org>
(C) 2013, Ryan MacDonald <ryan@r-fx.org>
inotifywait (C) 2007, Rohan McGovern <rohan@mcgovern.id.au>
This program may be freely redistributed under the terms of the GNU GPL v2

maldet(383): {quar} malware quarantined from 'scan/серв.php' to '/usr/local/maldetect/quar.
серв.php.28885'
maldet(383): {clean} restoring /usr/local/maldetect/quarantine/серв.php.28885 for clean
t
maldet(383): {clean} trying to clean scan/серв.php with gzbase64.inject.unclassified rule
maldet(383): {clean} rescanning scan/серв.php for malware hits
maldet(383): {clean} clean successful on scan/серв.php
```

► Рис. 6. Работа с карантинном *LMD*.

В завершение приведем пример использования *LMD* с *Inotify*:

```
$sudo maldet --monitor ~/scan
```

В данном случае будет осуществляться мониторинг каталога `scan` в домашнем каталоге текущего пользователя. А все изменения будут заноситься в файл `inotify_log`, который находится в каталоге `/usr/local/maldetect/inotify`.

Заключение

Linux Malware Detect ориентирован на выявление вредоносного программного обеспечения в режиме сканера. Его возможности основаны на специфических базах сигнатур, которые регулярно обновляются.

LMD прост в установке и позволяет осуществлять все базовые действия, характерные для данной разновидности ПО: сканирование (как в обычном, так и в фоновом режиме), лечение, помещение файлов с вредоносным содержимым в карантин и т. д. Кроме того, сканер дает возможность отправки подозрительных файлов на сервер своих разработчиков. А отличительной особенностью является поддержка мониторинга изменений тех или иных каталогов с помощью *Inotify*.

Программа имеет подробное руководство, конфигурационный файл хорошо документирован, а вывод в случае необходимости (например, при возникновении ошибки) снабжается подробной информацией. Таким образом, *LMD* отлично подойдет в качестве средства для повышения уровня безопасности вашего сервера. **LXF**

ССЫЛКИ

- 1 Статья в Википедии, посвященная обфускации <http://ru.wikipedia.org/wiki/Обфускация>
- 2 Официальный сайт Linux Malware Detect <http://www.rfxnetworks.com/projects/linux-malware-detect/>
- 3 Официальный сайт портала CLEAN MX <http://www.clean-mx.de/>
- 4 Сайт портала Malware Domain List <http://www.malwaredomainlist.com/>
- 5 RSS-канал, посвященный обновлениям сигнатур *LMD* <http://www.rfxn.com/api/lmd.php>

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 Избавление от неиспользуемых рабочих столов
- 2 Защиту портов VNC
- 3 Решение проблем со скриптами Bash
- 4 Установку дистрибутивов с DVD
- 5 Шпионаж за своими детьми
- 6 Хранение медиа-файлов

1 Уборка стола

На DVD LXF174 есть версия Linux Mint 15 с тремя рабочими столами и Unity. Я установил ее на свободный раздел своего жесткого диска и опробовал все рабочие столы.

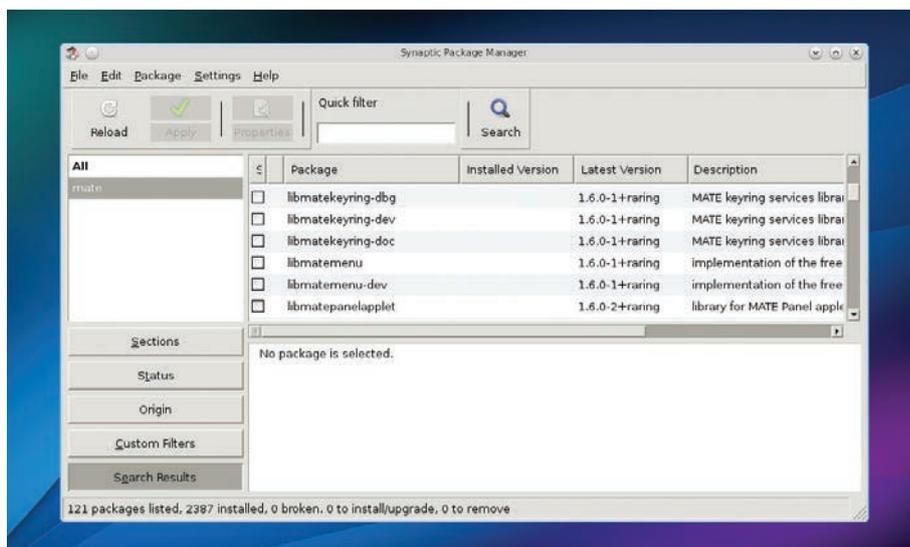
Мне больше всего нравится KDE, и я установил много дополнительных пакетов, в том числе, например, стиль Oхуген для *LibreOffice* и *Firefox*. Другие рабочие столы я хотел бы удалить, но не знаю, как сделать это безопасно, не повлияв на KDE. Не могли бы вы описать эту процедуру? Стоит ли воспользоваться *Synaptic*, и если да, то как сделать это без риска для KDE? И последний вопрос: нельзя ли поместить на один из будущих дисков последнюю версию дистрибутива Parted Magic? Последний раз, насколько я помню, был в LXF162 около года назад.

Питер Лонсдейл [Peter Lonsdale]

Удалить рабочие столы Mate, Cinnamon и Unity можно, но делать это следует из терминала командами `dpkg` и `apt-get`. В качестве альтернативного метода можно воспользоваться поиском *Synaptic* для нахождения всех установленных пакетов Mate, Cinnamon и Unity и удалить их, но таким образом Вы не только рискуете переустановить кисти рук, но и не избавитесь ото всех зависимостей.

Команда `dpkg --get-selections` выведет список всех установленных пакетов. Пропустив его через *Awk*, можно отфильтровать те, что соответствуют нужному нам имени, и выводить только актуальные имена пакетов. Затем мы передаем этот список `apt-get`, пользуясь параметром `purge` для удаления всех файлов, включая конфигурационные — если хотите сохранить их, пользуйтесь параметром `remove`:

```
sudo apt-get --just-print --auto-remove purge
$(dpkg --get-selections | awk '/^mate/ {print $1}')
sudo apt-get --just-print --auto-remove purge $(dpkg
--get-selections | awk '/^cinnamon/ {print $1}')
sudo apt-get --just-print --auto-remove purge
$(dpkg --get-selections | awk '/^unity/ {print $1}')
```



» *Synaptic* удобен для удаления одиночных пакетов, но для групповых операций лучше подойдет командная строка.

Как видите, команда запускается трижды (по разу на каждый рабочий стол, который нужно удалить). Можно объединить все три команды в одну, но это резко повышает вероятность опечаток или просто невнимательного прочтения вывода команды, что может вызвать проблемы. С параметром `--auto-remove` также удалятся все пакеты, которые установлены только как зависимости удаляемых Вами пакетов, а параметр `--just-print` выполняет «пустой прогон» команд: выводится только список пакетов — кандидатов на удаление, но пока они не удаляются. Проверьте, что среди удаляемых пакетов нет ничего ценного, и повторите все команды уже без параметра `--justprint`.

В порядке последней меры предосторожности выполните перезагрузку и обновление пакетов — либо из *Synaptic*, либо из терминала, раз уж он открыт:

```
sudo apt-get update
sudo apt-get upgrade
```

Эти команды должны переустановить все, что могло быть удалено, но все еще необходимо оставшимся у Вас программам. Я подумываю добавить скрипт для удаления рабочих столов в очередной дистрибутив с несколькими рабочими столами. Что касается Parted Magic, новая версия вышла в августе и есть на DVD LXF176.

2 Защита VNC

В Мой порт VNC проброшен через роутер, и я могу подключиться к настольному компьютеру, будучи вне пределов сети. Проблема в том, что как я недавно обнаружил, размер

файла `.xsessionerrors` достиг нескольких гигабайт. Файл заполнен попытками входа через VNC. Хорошая новость такова, что журнал делает то, что мне нужно — информирует меня об атаках. Но как можно остановить/предотвратить их?

Я попытался изменить порт IP, но тогда VNC, похоже, вообще перестал подключаться. Я попытался сделать перекрестный порт, чтобы в моей сети он остался по умолчанию, а вне ее был портом, который я задаю сам, но безуспешно. Затем я попытался пробросить выбранный мною порт и настроить сервер VNC (я попробовал *TightVNC* и *Vino*), но не смог подключиться с этими настройками.

Роб Джулиано [Rob Giuliano]

Если Вы открываете порт внешнему миру, люди норовят проникнуть через него. Вы попытались выбрать нестандартный порт, а это лишь прячет точку проникновения, но не закрывает ее. Причина Вашей неудачи, вероятно, в том, что для работы VNC нужно два открытых порта — по умолчанию, 5800 и 5900. Нужно изменить оба этих порта на сервере и на клиенте, а также настроить перенаправление на них на роутере. Это снизит количество попыток входа в систему, но не предотвратит их полностью, так как сканирование портов может определить, какие порты открыты и какие сервисы на них запущены.

Для предотвращения несанкционированного входа в систему, кроме выбора надежного пароля, есть и несколько других вариантов. Один из них — настроить виртуальную частную сеть,

VPN [описано в LXF158, Учебники, стр. 86]. Этот вариант требует кое-какой настройки, но когда она сделана, это не только безопасно (так как для аутентификации используются сертификаты, защищенные паролем), но и может использоваться для обращения к любым другим сервисам без открытия дополнительных портов на роутере.

Другой вариант — что-нибудь вроде *Fail2ban* (www.fail2ban.org). *Fail2ban* отслеживает содержимое лог-файлов на наличие неудачных попыток входа в систему. Когда их количество для определенного IP-адреса превышает заданную величину, программа добавляет правило брандмауэра, которое блокирует доступ с этого адреса. Если в течение какого-то времени попыток подключения с этого адреса не возникает, правило истекает. Так Вы получаете динамически обновляемый брандмауэр, который блокирует попытки подключения к Вашей системе с помощью любых сервисов. *Fail2ban* есть в репозиториях пакетов большинства дистрибутивов; наряду с подробной онлайн-документацией, у программы есть набор шаблонов конфигурации для популярных сервисов.

3 Слушай мою команду

В Linux я перешел сравнительно недавно, но многие годы пользовался программами для командной строки в Windows. Одна из возможностей терминала Windows (с помощью команды `take от jrssoft`) — отображение списка файлов, откуда затем выбираются те, над которыми нужно выполнить действие, т.е. возможность визуально выбрать файлы. Я не смог найти такой функции в *Bash*, поэтому написал несколько простых скриптов, обеспечивающих эти функции.

Эти скрипты находятся в моем домашнем каталоге, и я создал алиас, чтобы каждый раз не добавлять путь к этим скриптам в PATH. Все они записывают файл или имена файлов в текущий

каталог и затем работают с этими файлами. Знакомы ли вам какие-то программы, которые, так сказать, более отполированы по сравнению с моими самодельными скриптами *Bash*?

Роберт Соломон [Robert Solomon]

У нас нет места, чтобы привести Ваши скрипты, но у Вас интересный подход, и Вы хорошо поработали, создав решение, отвечающее Вашим нуждам. Что до расположения скриптов, их нужно разместить в `~/bin`, а не в домашнем каталоге. Этот каталог обычно содержится в переменной PATH, а если его нет, его легко добавить, и Вам не потребуются алиасы или полные пути. Другой вариант — определить операции, которые Вам нужны, как функции в своем профиле, добавив их в `~/bash_profile`:

```
makelist() {
    командны будут здесь
    и здесь
}
```

Эти функции доступны любой оболочке, как если бы они были встроенными командами или скриптами из PATH.

Однако похоже, что на самом деле Вам нужен какой-то файловый менеджер для терминала, и их множество. Самый популярный — *Midnight Commander* (www.midnight-commander.org), который, возможно, уже установлен в Вашей системе. Если нет, Вы точно найдете его в своем менеджере пакетов, обычно под именем `mc`. *Midnight Commander* содержит стандартный двухпанельный интерфейс файлового менеджера, но в терминале и под управлением клавиатуры. Функциональные клавиши выполняют различные действия. Больше всего Вам пригодятся F2 для выполнения действий над выбранными файлами, F9 для переключения в меню и, конечно, F1 для вызова справки. Документация в справке достаточно подробна,

Терминалы и суперпользователи

Мы часто предлагаем в качестве решения проблемы ввести те или иные команды в терминале. Хотя обычно то же самое можно сделать с помощью графических утилит дистрибутива, такие решения будут слишком конкретными (будут зависеть от дистрибутива). Команды в терминале более гибкие и — самое главное — ими можно пользоваться во всех дистрибутивах. Команды настройки системы часто нужно выполнять от имени суперпользователя, называемого также `root`. Существует два основных способа это делать, в зависимости от используемого дистрибутива. Во многих дистрибутивах, особенно в Ubuntu и его производных, перед командой можно написать `sudo` — при этом будет запрошен пароль пользователя, и ему будут предоставлены привилегии `root` только на время выполнения команды. В других дистрибутивах применяется команда `su`, для использования которой требуется ввести пароль `root` и которая предоставляет полный доступ `root` до того момента, пока вы не наберете `logout`. Если в вашем дистрибутиве используется `su`, запустите ее один раз и выполняйте любые заданные команды без предшествующей `sudo`.

правда, не по части Вашего вопроса. Для выбора файлов для последующих действий нажмите Insert. Наряду с командами в меню с помощью F2-@ можно передать файлы любой другой команде или запускать команды напрямую из встроенной командной строки.



Коротко про...

Тор

Случалось ли вам недоумевать, почему ваш компьютер вдруг стал тормозить, зависать или вгонять в клин жесткий диск? Если это произойдет снова, запустите в терминале команду `top`. Эта программа компактно отображает множество информации. По умолчанию это системная статистика — время работы системы, число пользователей, количество работающих задач и т.д. Затем она показывает общую занятость процессора, поделенного между программами пользователя (`us`), системными операциями (`sy`), процессами с измененным приоритетом (`ni`), простым (`id`) и ожидающими (`wa`) процессами. «Простой», очевидно, относится не к процессам, а показывает, сколько свободных процессорных ресурсов у вас

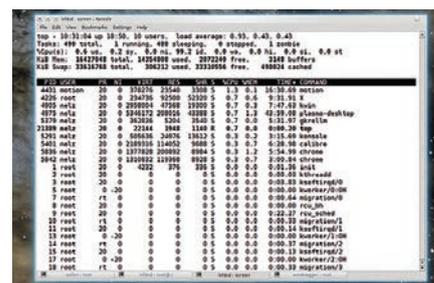
есть. Ожидающие процессы — процессы, захватившие ресурсы в процессе ожидания чего-то еще, обычно ввода/вывода.

Ниже располагается список работающих задач, отсортированных по потреблению процессора. Благодаря этому списку вам сразу станет ясно, какая из них съедает все циклы процессора. Для изменения сортировки, например, для сортировки по потреблению памяти, нажмите F.

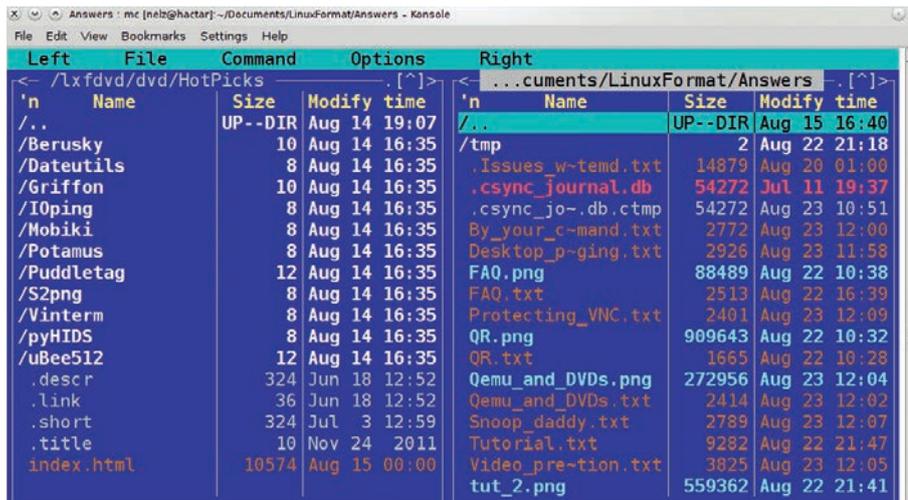
Процессы не просто отображаются — ими также можно управлять. Нажмите K, чтобы убить процесс, или R, чтобы изменить его приоритет — увеличение значения `nice` [вежливость] делает процесс менее «жадным».

Чем вежливее программа, тем чаще она говорит другим «только после вас». Для уничтожения процесса или изменения приоритета необходим идентификатор процесса (PID), указанный в левой колонке. Опечатка здесь может привести к печальным последствиям, но если

`top` запущен в терминале X, можно дважды щелкнуть на PID необходимого процесса и нажать среднюю кнопку мыши, чтобы вставить его. У программы есть множество других опций; для доступа к большинству из них достаточно нажатия одной клавиши. Самая важная клавиша — H, отображающая интерактивную справку.



» Top показывает, какая часть ценных ресурсов процессора и памяти используется.



➤ **Midnight Commander** добавляет мощь графического файлового менеджера в ваш терминал или консоль.

4 Qemu и DVD

В Я много лет выписываю Linux Format и накопила кучу установочных DVD. Недавно я заинтересовалась виртуальными машинами и захотела установить некоторые дистрибутивы с DVD. Но как? ISO-образов нет, и скриптов для их создания тоже нет.

Можно ли установить один дистрибутив с DVD в Qemu с командной строки без необходимости загружать ISO? Я хотела бы запустить DVD в обычном сеансе, установить дистрибутив с DVD, вынуть DVD из лотка — и готово.

Шэрон [Sharon]

О Дистрибутивы можно установить прямо с DVD (так я сам проверяю дистрибутивы с LXF DVD). Вместо ISO-образа укажите имя своего привода DVD:

```
qemu hd.img -cdrom /dev/dvd -boot d
```

Однако работать с ISO-образами намного быстрее, и время, затраченное на конвертацию DVD в ISO-образ, более чем окупится, если Вы будете загружаться с одного диска несколько раз. На DVD есть скрипт *Usb2iso.sh*, предназначенный, как следует из названия, для переноса содержимого DVD на USB-брелок.

Для этого он создает так называемый гибридный ISO-образ, который можно скопировать на брелок, чтобы затем загрузиться с него, и который работает как обычный ISO-образ. Этот скрипт помещается на наши DVD уже несколько лет. Если запустить с DVD *Dvd2usb.sh* и выбрать создание файла (вместо копирования на USB-брелок), программа создаст в текущем каталоге файл образа, пригодный для использования в Qemu или любом другом эмуляторе:

```
bash /media/dvd/dvd2usb.sh
qemu hd.img -cdrom LXF DVD177.img -boot d
```

Обратите внимание, что хотя при копировании на USB-брелок этот скрипт нужно запускать с **sudo**, при создании образа использовать **sudo** не нужно и нежелательно, иначе владельцем файла образа станет суперпользователь-root.

Если Вы хотите создать образы отдельных дистрибутивов с DVD, это возможно для нескольких

последних номеров при посредстве программы *MakeISO*, которая находится в большинстве каталогов с дистрибутивами на DVD. *MakeISO* нужно запускать из того места, где она находится; если попробовать скопировать ее куда-то еще, при запуске она выдаст ошибку. До последнего месяца *MakeISO* не работала с Debian, так как там не было *Mkisofs* — но теперь мы это обошли. Например, чтобы создать ISO-образ *Perrepmint4* с последнего DVD, выполните команду:

```
bash /media/dvd/Peppermint4/MakeISO
```

5 Бдительный папа

В Пару месяцев назад я застал свою дочь за поиском непристойностей в Google и отрезал ей Интернет. В конце концов она получила его обратно (но уже с несколькими фильтрами). Можно ли как-то заставить ее компьютер отправлять мне историю просмотра в браузере каждый раз, когда она входит в систему? Так я смог бы следить за ней, не давая ей об этом знать. Было бы хорошо узнать ее поисковые запросы в Google, потому что именно за этим я ее и поймал в прошлый раз.

Heiowge, С форумов

О Конечно, можно создать скрипт для отправки файлов истории браузера при ее входе в систему, поместив команду в *~/.profilerc*. Нечто вроде

```
cat ~/.mozilla/firefox/default.ZZZ/history.dat | mail -s "Firefox history" you@your.email
```

Этот способ имеет некоторые недостатки, и его легко обойти. Затеявая что-то неприличное, она может воспользоваться другим браузером, переключиться в приватный режим или удалить файлы истории браузера. Более надежный подход — воспользоваться прокси-сервером. Полноценный прокси-сервер вроде *Squid* Вам ни к чему — достаточно такого, который просто принимает запросы и пропускает их, записывая информацию в лог-файл, например, *Tinyproxy* (www.banu.com/tinyproxy). Установите его на свой компьютер, который, очевидно, должен быть включен, когда Ваша дочь пользуется Интернетом, и измените

конфигурационный файл */etc/tinyproxy.conf* в соответствии со своими потребностями. В частности, Вам нужно убедиться, что задан параметр *LogFile* и что *LogLevel* установлен в "Connect".

Теперь обеспечьте, чтобы компьютер Вашей дочери отправлял все запросы через прокси — для этого нужно задать настройки прокси в каждом установленном браузере. В них нужно указать IP-адрес Вашего компьютера и номер порта, указанный в *tinyproxy.conf*, по умолчанию 8888. Включите использование прокси для всех сервисов, и все запросы браузера будут проходить через Ваш компьютер и фиксироваться в лог-файлах. После этого в любой момент Вы сможете заглянуть в лог-файл.

Вы говорите, что не хотите, чтобы дочь об этом знала, хотя это могло бы улучшить ее поведение; но рано или поздно узнать ей придется, если уж Вы застанете ее за чем-то плохим. Поэтому Вам нужно гарантировать, что она не сможет отключить прокси. Можно воспользоваться подходом грубой силы и сделать соответствующие конфигурационные файлы, например, *~/.mozilla/firefox/default.ZZZ/prefs.js*, доступными только для чтения, хотя решительному ребенку это не помешает создать новый профиль. В качестве альтернативы можно настроить блокировку исходящих запросов на порты 80 и 443 (HTTP и HTTPS) с данного конкретного компьютера, вынудив их проходить через прокси-сервер. *Tinyproxy* также поддерживает фильтрацию, и когда он заработает, Вы можете переместить свои правила фильтрации сюда, где до них никто не доберется.

6 Хранение видео

В Я начал переносить домашнее видео с касет на DVD — в основном потому, что забарахлила моя видеокамера. Я хотел бы узнать, можно ли защитить файлы RAW контрольной суммой, чтобы при небольшом повреждении файла его можно было восстановить? Конечно, для всех файлов у меня есть резервные копии, но я хотел бы добавить еще немного уверенности в том, что мои ценные воспоминания сохранятся в целости и в будущем я смогу преобразовать их в следующий формат мультимедиа.

Ричард Стрит [Richard Street]

О Создать контрольную сумму для файла, чтобы определить, что он был поврежден, очень просто. Например, следующей командой можно сгенерировать контрольную сумму MD5 для одного файла:

```
md5sum filename
```

Эту команду можно также объединить с **find**, чтобы создать файл, содержащий контрольные суммы всех файлов в каталоге:

```
md5sum >MD5SUMS $(find dirname -type f)
```

Список всех контрольных сумм сохранится в файле MD5SUMS, который можно проверить командой

```
md5sum --check MD5SUMS
```

Но вопрос о том, как восстановить данные, если проверка контрольной суммы показала ошибку, по-прежнему остается открытым.



» С любого из ваших DVD можно загрузиться в виртуальной машине — или напрямую, или создав ISO-образ.

Очевидный ответ — пользоваться резервными копиями, которые Вы создаете, и также проверять их командой `md5sum --check` на соответствие оригиналам. Если Вы храните копии на жестких дисках, где места много и оно дешево, это, пожалуй, лучшая альтернатива, особенно если две копии находятся физически в разных местах. Есть способы создания дополнительных данных для восстановления ошибок, чтобы при повреждении файла его можно было восстановить, но эти данные обычно хранятся на том же носителе и потому подвержены тем же рискам,

что и сами данные. Популярной программой для таких целей — в основном для борьбы с ошибками передачи, повреждающими загруженные файлы — была *par2*, но она не поддерживается уже несколько лет, и я не уверен, что она будет работать нормально сейчас или, что гораздо важнее, в будущем, когда она Вам на самом деле понадобится.

Если Вы планируете хранить резервные копии на оптических дисках, DVD или BD, *DVDisaster* (<http://dvdaster.net>) — возможно, именно то, что Вам нужно. Она хранит информацию для восстановления ошибок на диске вместе с исходными файлами за счет примерно 15% дополнительного места на диске. Другой вариант — хранить эти данные в отдельном файле. ISO-образ своих файлов можно создать в любимой программе для создания образов компакт-дисков. После этого загрузите его в *DVDisaster*. На вкладке Preferences > Error Correction [Настройки > Коррекция ошибок] выберите либо Error Correction > File [Коррекция ошибок > Файл], либо Augmented Image [Наращиваемый образ], для хранения информации об ошибках в отдельном файле или в том же файле соответственно, затем нажмите Create [Создать]. Во втором случае будет создан ISO-образ большего размера, содержащий все данные для восстановления ошибок, которые затем можно восстановить в *DVDisaster*. **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 нужно знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства.

Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (<http://hardinfo.berlios.de/>) — она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файле, который вы сможете приложить к своему письму.

Не уступающий в удобстве альтернативный вариант — *lshw* (<http://ezix.org/project/wiki/HardwareLiSter>). Одна из указанных программ непременно должна быть включена в ваш дистрибутив (а иногда и обе).

Если вы не хотите или не можете их установить, выполните следующие команды в терминале от имени root и приложите файл `system.txt` к письму. Это здорово поможет диагностике.

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```



Часто задаваемые вопросы

Rsync и Unison

» Что это за программа *rsync*, о которой все болтают?

Это способ синхронизации содержимого двух каталогов, гарантирующий, что один каталог будет точной копией другого.

» Нельзя ли воспользоваться *cp*?

Можно, но *cp* копирует все. *Rsync* копирует только те файлы, которые отличаются. Если изменились большие файлы, копируются только измененные фрагменты.

» Как ею воспользоваться?

После выполнения команды `rsync --archive --delete /path/to/source/ /path/to/dest/`

второй каталог будет содержать точную копию первого. Параметр `--delete` удаляет файлы, которых нет в первом каталоге, а `--archive` копирует все права доступа и временные отметки файлов. С *Rsync* важно указывать завершающие слэши; они показывают, что вы хотите синхронизировать содержимое каталогов.

Если их опустить, это может привести к тому, что один каталог скопируется в другой.

» А если мне надо синхронизироваться с каталогом на другом компьютере?

Если у вас есть SSH-доступ к другому компьютеру, можете воспользоваться командой

```
rsync --archive --delete /my/local/site/ hostname:/path/to/site/
```

Когда в одном или в обоих путях указано имя хоста, *Rsync* использует для передачи файлов удаленную оболочку. По умолчанию это будет SSH.

» Что если я изменил файлы на двух компьютерах и хочу их синхронизировать?

Нужно нечто другое, более приспособленное для этой задачи, например *Unison* (www.cis.upenn.edu/~bcpcierce/unison). Программа использует *Rsync* и SSH для передачи файлов, поэтому все их пре-

имущества по-прежнему в вашем распоряжении, но она предназначена для двунаправленной синхронизации. Если вы изменили какие-то файлы на одном компьютере и какие-то на другом, она гарантирует, что на каждом компьютере будет последняя версия файлов.

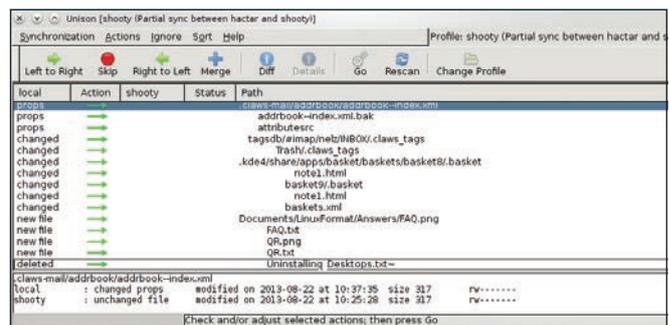
» Но если я изменю один и тот же файл на обоих компьютерах?

Программа не может прочесть ваши мысли и узнать, какую версию сохранить, но может сказать, что изменены обе копии, и спросить, которой воспользоваться. *Unison*

отслеживает все изменения, и будет знать, что файл был изменен на обоих компьютерах с момента ее последнего запуска.

» И, конечно, у командной строки десятки параметров?

Да, но можно выбирать файлы и в графическом интерфейсе. Графический интерфейс запускается только на компьютере, инициирующем передачу данных, поэтому с его помощью можно синхронизироваться с удаленным компьютером, если вы предварительно установили там *Unison*.



» Держите ноутбук и настольный компьютер в *Unison*.



LXF HotPicks



Майк Сондерс
Издавив самые недостижимые и укромные уголки Интернета, Майк точно знает, где таятся главные сокровища открытого кода.

CherryTree » Lightspark » Lynx » ClassicMenu » Jpgind » 4MLinux
» Виджет Equake » Savage Wheels » Mr Rescue » Clzip » Pragma

Приложение для заметок

CherryTree

Версия 0.30.1 Сайт www.giuspen.com/cherrytree

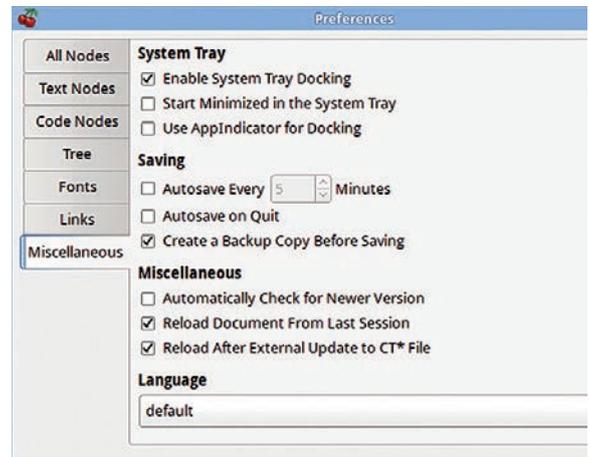
Мы отнюдь не бросаемся словом «выдающийся», однако на каждом шагу встречаем блистательные образчики ПО, заслуживающие подобного отзыва. Еще более впечатляет, что в данном случае это приложение для заметок. Возможно, вы думаете: да что такое «выдающееся приложение для заметок»? Не слишком ли громкая похвала для программы, которая ничего выдающегося не делает? В данном случае, *CherryTree*

включает абсолютно все мыслимые функции — и они на диво хорошо реализованы.

CherryTree организует данные в иерархическое древо узлов, и вы можете хранить свою информацию по категориям и создавать ссылки между ними. Он включает редактор rich-text с обычными опциями форматирования (цвета, жирный шрифт, курсив), плюс выделение синтаксиса для многих языков программирования и возможность размещать изображения внутри заметок.

А добавок — маркированные и нумерованные списки, таблицы и гиперссылки на веб-страницы. Ну, а панель

«Панель редактирования — прямо-таки текстовый редактор.»



» Через Edit > Preferences можно настроить автосохранение и поместить значок программы в область уведомлений.

редактирования — прямо-таки настоящий небольшой текстовый редактор: ее способности впечатляют.

И это мы еще только начали! Если у вас уже есть заметки, сохраняемые в другой программе, их легко импортировать в *CherryTree*: форматы *Zim*, *TomBoy*, *Basket*, *NoteCase* и многие другие здесь поддерживаются. *CherryTree* умеет сохранять собственные данные в базе данных *SQLite*, а если вы хотите изменить свои заметки в других программах, можете выбрать вместо нее XML. В обоих случаях есть возможность поставить пароль. Кроме того, есть полноценный набор опций для управления и сортировки узлов, возможность экспорта в plain text, HTML или PDF, клавиши быстрого запуска практически для любой задачи и возможность поиска по регулярным выражениям.

При таком изобилии функций вам может показаться трудным приступить к работе, но справочник программы доступен в формате *CherryTree* (скачайте файл СТВ с сайта), и вы увидите примеры использования всего, что здесь есть. А поскольку она написана на Python с основными зависимостями *PyGtk*, *Enchant* и *gtkSourceView*, к работе приступить легко — просто выполните скрипт **cherrytree** в полученной директории после распаковки tar-архива.

Исследуем интерфейс CherryTree

Панель инструментов

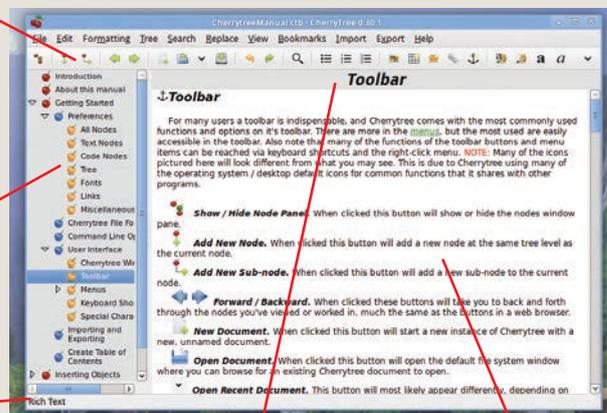
Используйте эти кнопки для добавления узлов и форматирования текста — у них у всех есть подсказки. Если все они не помещаются на вашем экране, щелкните по стрелке вниз справа.

Древо

Каждый пункт в этом списке представляет собой узел, и вы можете добавлять дочерние узлы к уже существующим, создавая иерархическую структуру.

Режим редактирования

В большинстве случаев это будет Rich Text (т.е. с опциями форматирования), но если вы редактируете код, то язык, используемый для выделения синтаксиса, будет отображаться здесь.



Название

Показывает название текущего узла — нажмите F2, чтобы изменить имя (и язык программирования, при необходимости).

Редактор

Панель редактирования. Нажмите F9, чтобы отключить левое древо и освободить место.

Flash-плеер с открытым кодом

Lightspark

Версия 0.7.2 Сайт <http://lightspark.github.io>

Вам наверняка знакомо это чувство: браузер застопорился, потребление CPU зашкаливает, и у всего этого один виновник. Завис проприетарный плагин Flash, уже в седьмой раз за сегодняшний день. Редкий пользователь Linux является фанатом Flash, и хотя многие из нас стараются по возможности избегать его, ряд популярных сайтов по-прежнему его применяет. Будем справедливы: не во всех проблемах с Flash виноват Adobe или его плагин с закрытым кодом; просто некоторые сайты заходят слишком далеко, одновременно запуская неумеренное количество Flash-апплетов.

Достойное расширение браузера для блокировки Flash справляется с этой проблемой, но вам все равно навязывают капризный, пожирающий память плагин, который нельзя изменить. Было несколько попыток реализации Flash-плеера с открытым кодом, и *Lightspark* — одна из самых многообещающих. Она доступна как в виде утилиты командной строки (для

воспроизведения файлов SWF), так и в виде плагина браузера — если вы используете дистрибутив на базе Ubuntu, первую вы найдете в пакете *lightspark*, а второй — в виде *browser-plugin-lightspark*. Если у вас другой дистрибутив, или вы решите скомпилировать *Lightspark* из исходника, список зависимостей ищите в файле **Readme**. В любом случае сначала придется удалить плагин Adobe, дабы избежать конфликтов.

Основное внимание *Lightspark* уделяет тому, чтобы заставить работать YouTube — однако случается всякое. В наших тестах некоторые видео воспроизводились идеально, у некоторых были проблемы со звуком (самая распространенная проблема — крайне тихий звук), а некоторые вообще отказывались работать, выдавая со-

«Основное внимание *Lightspark* уделяет работе YouTube.»



» В основном YouTube работает хорошо, но у Flash-игр часто бывают трудности — см. «слепые» области на картинке.

общение «неподдерживаемый файл Flash [unsupported Flash file]». Рады сообщить, что большинство видео все же воспроизводилось нормально: мы протестировали 20 случайно выбранных клипов с YouTube, и 17 из них работали без всяких аудио- или видеопроблем. Разработчики говорят, что *Lightspark* «вполне нормально поддерживает YouTube для повседневного использования», и по нашему мнению, эта оценка вполне справедлива.

Однако с более интерактивным контентом *Lightspark* сражается очень долго, как мы обнаружили, взявшись попробовать разные онлайн-игры на Flash. Впрочем, до версии 1.0 еще предстоит долгий путь. Заглядывайте на страницу wiki на <http://tinyurl.com/77xuzeg>.

Web-браузер

Lynx

Версия 2.8.8-dev-16 Сайт <http://lynx.isc.org>

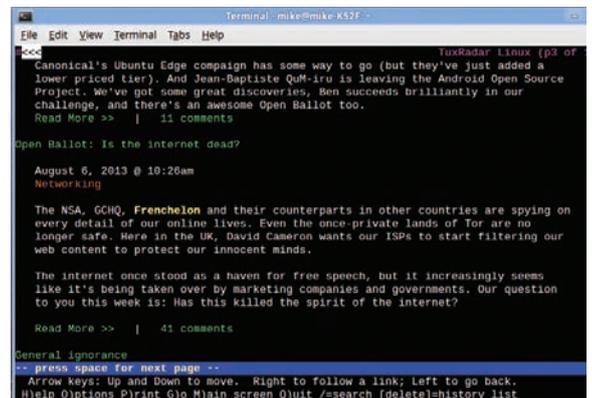
Если вы уже давно слушаете TuxRadar Podcast (а если нет, то почему?), то вы, возможно, помните наш конкурс по командной строке (Сезон 2, Серии 1 и 2). Мы тогда целую неделю не имели права пользоваться ничем, кроме текстовых программ — ни *Firefox*, ни *LibreOffice*, ни *Amarok*, ничего такого. Найти в командной строке замену наших любимых программ типа наведи-и-щелкни оказалось совсем не сложно, и мы обнаружили, что жизнь без растрового GUI вовсе не так ужасна, как воображают некоторые.

Навигация по Сети в текстовом терминале сначала кажется странной, но есть несколько браузеров, весьма достойно справляющихся с этой работой. Самый старый из них — *Lynx*, абсолютная древность с точки зрения ПО, поскольку он появился аж в 1992 году. И мы, конечно, радостно заулыбались, увидев, что программа все еще фурычит и даже получает обновления; пусть среди них и нет

блестящих новых функций, но она остается совместима с web-стандартами и разновидностями Unix/Linux.

Мы попробовали релиз dev-16 того, что собирается стать *Lynx* 2.8.8. Его зависимости минимальны: для компиляции нужно всего лишь наличие библиотеки *ncurses*. Запустив его, вы попадете на web-страницу *Lynx*, и это — отличное место для изучения всех привязок клавиш. Ссылки отображаются зеленым, а выделенная на текущий момент ссылка — желтым; используйте кнопки со стрелками, чтобы выбрать другие ссылки, и Enter, чтобы по ним перейти. Нажав G, вы сможете ввести новый URL, а Backspace перенесет вас в журнал [History].

«Как резервный вариант, *Lynx* будет очень удобен.»



» Рендеринг раскладки *Lynx* не столь хорош, как у *Links* (другой текстовый браузер), но все же впечатляет.

По умолчанию *Lynx* спрашивает вас, хотите ли вы принимать cookies, при посещении любого сайта, который их использует, что может начать раздражать. Нажмите O, чтобы вывести экран опций, где можно изменить политику cookie; затем выберите Save Options To Disk > Accept Changes [Сохранить параметры на диске > Принять изменения].

Конечно, по количеству функций *Lynx* и рядом не стоял с *Firefox* или *Chrome*, но как резервный вариант он будет очень удобен. Также он хорош для дорогих мобильных широкополосных соединений: просмотр в текстовом режиме значительно сокращает потребление трафика.

Традиционное меню приложений

ClassicMenu

Версия 0.08 Сайт <http://bit.ly/18oxFph>

Нравится вам это или нет, но интерфейс Ubuntu Unity никуда не денется. Многие фанаты Ubuntu приуныли, когда дистрибутив переключился с довольно простого Gnome на совершенно новый рабочий стол, а некоторые вообще сменили дистрибутив. Другие научились любить Unity, да со временем он и улучшился; и в любом случае, попытка Canonical создать стабильный интерфейс для работы на нескольких платформах заслуживает внимания.

Одной из главных проблем с Unity была программа запуска приложений. К счастью, хотя Unity отличается весьма жесткими требованиями, он открыт для таких сторонних улучшений, как *ClassicMenu*. Эта программка предоставит апплет области уведомлений, выдающий при щелчке по нему знакомое меню программы запуска, где приложения организованы по категориям. Она написана на Python, и для установки понадобятся зависимости *python-gmenu*, *python-appindicator*, *python-gtk2* и *python-gobject*. Устанавливается

ClassicMenu прямо из его единственного Deb-файла; альтернатива — добавить его PPA в источники своих программ. После установки выйдите из своей сессии Unity и снова войдите, и вы должны увидите индикатор *ClassicMenu* справа вверху на рабочем столе (это мелкий значок с тремя серыми треугольниками, уставленными по вертикали). Если он не появился, загрузите его вручную из терминала, введя **classicmenu-indicator**. Апплет не привязан к Unity и работает и в других рабочих столах и менеджерах окон с поддержкой области уведомлений — если вы захотите украсить свой минималистский менеджер окон достойным меню, он может помочь.

ClassicMenu генерирует свой список программ на основании содержимого `/usr/share/applications`, и незамедлительно



Вот *ClassicMenu*, работающее под Xfce — оно использует тему GTK среды рабочего стола.

откликается на перемены в этой директории; вам не потребуется перезапускать *ClassicMenu*, когда вы добавите очередное приложение. Чтобы настроить само *ClassicMenu*, загляните в `.config/classicmenu-indicator` в своей домашней директории — если там нет файла `config`, создайте его на основе шаблона с сайта программы. Через этот файл можно производить некоторые реформы: скажем, менять вид и размер главного значка.

«К счастью, Unity открыт для таких сторонних улучшений.»

Генератор web-галереи

Jpgind

Версия 0.87 Сайт <http://bit.ly/1agzquk>

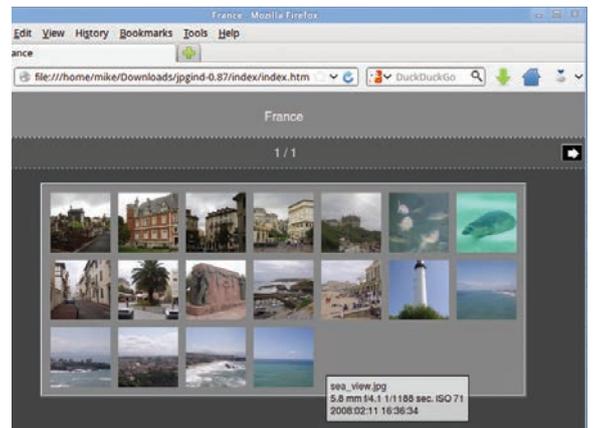
Этот смешной старый мир... В конце 1990-х, когда был бум доступа к Интернету среди домашних пользователей, мы все восторгались децентрализованной природой Интернета. В нашем 2013 году ситуация иная: большинство использует достаточно скромный набор сервисов из Google и Facebook, и если те хоть на пару минут отказывают, люди прямо-таки хватаются за оружие.

Итак, есть еще что сказать о размещении собственных данных и о большей независимости вашего существования, особенно при таком масштабе краж информации и шпионства. Одно из популярнейших применений персонального web-пространства — возможность делиться фотографиями; и хотя большинство умников умеют по-быстрому сваять HTML для примитивной галереи изображений, все же лучше воспользоваться инструментом, специально созданным для этого.

Jpgind берет директорию изображений и генерирует приятные web-галереи с эскизами и подсказками. Процесс его компиляции довольно странный: введите **make**, и он скачает разные зависимости библиотек изображений из Интернета, запустит скрипты настройки и скомпилирует их статически связанными с финальным бинарником. В принципе здорово скачать все одной-единственной командой, но если у вас не хватает некоего инструмента, вам придется в подробностях изучить сообщения об ошибках. Зато по завершении компиляции создание галереи становится сущим пустяком:

```
./jpgind my_pics/
```

«Есть еще что сказать о размещении собственных данных.»



Jpgind работает в первую очередь с изображениями JPEG, но умеет обрабатывать и изображения RAW.

Jpgind генерирует новую директорию (в текущей) под названием `index`. Откройте внутри нее `index.html`, и вы увидите галерею на основе эскизов на каждый скриншот. Поводив курсором у эскиза, вы увидите данные EXIF, а щелкнув по эскизу, получите его полномерное изображение.

Jpgind достаточно полезен и в базовой форме, но может также похвастаться массой опций командной строки для элегантной индивидуальной настройки результата. Это впечатляюще гибкий инструмент; мы будем применять его ради дистанцирования от неумолимых сил Интернета.

Мини-дистрибутив с уклоном в мультимедиа

4MLinux Media Edition

Версия 7.0 Сайт <http://bit.ly/1agAfTT>

Мы уже рассказывали о двух версиях 4MLinux в недавних HotPicks: Antivirus Live CD в LXF174, а затем — TheSSS в LXF175. Это семейство мини-дистрибутивов продолжает расширяться, и теперь у нас есть Media Edition, которое предлагает подборку программ для воспроизведения, записи и нарезки аудио и видео. Его ISO-образ весит чуть больше 100 МБ и совместим с *UN-etchbootin*: можно записать его на USB-брелок и загружать оттуда (если вы не хотите возиться с CD-R).

4MLinux Media Edition требует ОЗУ от 512 МБ, благодаря чему доступен для более старого оборудования, хотя вряд ли вы добьетесь достойных результатов в декодировании HD-видео без приличного CPU и видеокарты. При первой загрузке вам предложат задать пароль root, и вы попадете в текстовое окно приглашения — уж не знаем, почему разработчики не включили графический менеджер входа, но просто войдите как root и введите **startx**, чтобы вывести GUI.

GUI легкий и минималистский: он состоит из менеджера окон *JWM* и симпатичной программы запуска *wbar*, увеличивающей значки. Верхняя панель предоставляет доступ к основным программам — это *SMPlayer 0.8.5* (усиленный моментальным SVN-снимком *MPlayer*), *QupZilla 1.4.3* (web-браузер на базе *Qt*) и браузер изображений *ZGV*. Предустановлено множество библиотек кодеков, и масса дополнительных инструментов доступна в меню внизу слева.

Удобно, что программы сгруппированы в соответствии с тем, что пользователю может понадобиться сделать: так, за подменю **Let's Play** [Воспроизведем] следует **Let's Rip** [Добудем с диска]. Есть также несколько ссылок меню, которые помогут скачивать добавочные программы.



➤ Весь дистрибутив влезает на 104 МБ, так что количество имеющихся в нем программ впечатляет.

В общем-то, 4MLinux Media Edition не самый интересный или богатый функциями дистрибутив для мультимедиа, но он компактный и аккуратный, и если вы знакомы с другими членами семейства 4MLinux, с освоением этого дистрибутива у вас не возникнет проблем.

«Это семейство мини-дистрибутивов продолжает расширяться.»

Монитор землетрясений

Виджет Equake

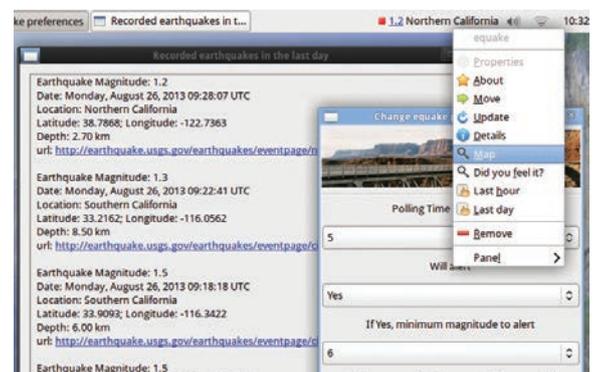
Версия 1.3.2 Сайт www.e-quake.org

Специально для наших читателей, живущих в сейсмоопасных зонах, а также для всех, кто интересуется поведением нашего маленького голубого шарика! Что ни день, в мире происходят сотни землетрясений, и большинство из них столь незначительны, что люди их не замечают, а часть происходит в отдаленных зонах, где от них некому пострадать. Однако периодически происходит нечто масштабное, и тогда мы наблюдаем жуткую картину разрушений. *Equake* — апплет для Gnome и *Xfce*, который отображает сводку землетрясений, недавно произошедших в мире, и дает ссылки на более подробную информацию.

Мы скомпилировали версию *Xfce* из исходника на Xubuntu 13.04: найти две из зависимостей не представило труда (*libxfce4util* и *libxfcegui4*), но у нас возникли сложности с *libxfce4panel*. Оказалось, что в дистрибутивах на основе Debian/Ubuntu

пакет именно этой зависимости имеет другое имя, и чтобы получить нужные файлы, нужно установить *xfce4-panel-dev*. Все установив, скомпилируйте апплет обычной процедурой **.configure, make** и **make install** (последний шаг — от имени root).

Чтобы добавить *Equake* в вашу панель *Xfce*, щелкните правой кнопкой по пустой области, перейдите в подменю **Panel** и выберите **Add New Items**. Затем найдите их в списке — или, если у вас много апплетов, используйте окно поиска. *Equake* покажет масштаб и место самых последних крупных землетрясений, а подсказки отобразят дополнительную информацию, и по щелчку правой кнопкой на апплете



➤ Похоже, Калифорнию сейчас потряхивает...

появится меню с дальнейшими ссылками: например, посмотрите на место землетрясения в Google Maps, или выведите список землетрясений за последний час или день.

Equake черпает данные в программе Геологического наблюдения США за угрозами землетрясений. Просто установите апплет в свою панель: видя, как он обновляется каждые 20 минут, вы вьязь ощутите активность нашей милой планеты. Вы поймаете себя на постоянном поглядывании на него чисто из любопытства. А жителям зоны повышенного риска шанс увидеть менее значительные землетрясения поблизости от себя даст время перебраться в место побезопаснее...

«Вы вьязь ощутите активность нашей милой планеты.»

HotGames Развлекательные приложения

Аркада с автомобильными авариями

Savage Wheels

Версия 1.5 Сайт <http://bit.ly/1dDhU3W>

Что за гонки без риска от души шархануть свою машину?! Нас удивляют эти разрекламированные высокобюджетные и недешевые гонки на современных консолях типа PS3 и Xbox 360, где, несмотря на превосходную графику и блестящие эффекты, ничего не происходит, если врезаться на машине в стену — в крайнем случае, какой-то малоубедительный звук. Допустим, не все гоночные игры обязаны быть правдоподобным автоимитатором, но отсутствие реализма в столкновениях отталкивает нас.

К счастью, *Savage Wheels* целиком посвящена столкновениям — в частности, с соперниками. Это не имитатор вождения в традиционном смысле — здесь нет кольцевых трасс или сложных дорожных препятствий. Вместо этого вы оказываетесь на арене вместе с несколькими

другими машинами, и ваша цель — уничтожить их. Стрелки вверх и вниз — акселератор и реверс, а клавиши вправо и влево управляют машиной. Внизу экрана — индикатор здоровья каждого участника: по мере повреждения вашей машины он уменьшается, а когда достигнет нуля, ваша машина взорвется.

Чтобы уничтожить другие машины, просто въезжайте в них. Хорошее столкновение вызовет массивные повреждения, и вы даже соберете бонусы. В игру включены разные режимы (доступные через едва различимое меню, встроенное в картинку шины на стартовом экране —

«Вы на арене; ваша цель — уничтожить другие машины.»



► **Выбирайте из нескольких машин — у каждой своя мощь и прочность.**

это на случай, если вы его пропустили), включая режим для одного и двух игроков. А уровень сложности настраивается.

Небольшая подсказка по установке: самый простой способ приступить к игре — скачать `savagewheels-1.5.0-Linux-x86.sh` или `savagewheels-1.5.0-Linux-x64.sh` и запустить его (например, `sh savagewheels-1.5.0-Linux-x86.sh`). У вас спросят, хотите ли вы установить ее в отдельную директорию, и как только файлы скопируются, прыгивайте в машинку и начинайте играть, используя `.run.sh`.

Платформенная игра

Mr Rescue

Версия 1.02 Сайт <http://bit.ly/175BE1p>

Следует признать, что многочасовой просмотр *Fireman Sam* [«Сэм-пожарный» — детский анимационный сериал, — прим. пер.] сильно искажил наши взгляды на работу пожарных. Мы выросли в убеждении, что пожарная служба существует исключительно для розыска пропавших драгоценностей, ремонта шин и снятия у людей кастрюль с головы. Куда более серьезная (и опасная) суть работы пожарного стала нам понятна намного позднее. Спасибо, детское ТВ.

В общем, так: *Mr Rescue* — это платформенная игра 2D в стиле ретро, которая загонит вас в шкуру пожарного. Она создана с помощью LÖVE (www.love2d.org), весьма впечатляющей среды для создания игр с помощью языка программирования Lua. После установки просто распакуйте файл `mrrescue-1.02b-love.zip`, перейдите в полученную директорию

и введите `love mrrescue.love`, чтобы запустить игру.

Вы сразу же увидите — и услышите — 8-битную атмосферу игры, с шумной фоновой музыкой и угловатой пиксельной графикой. Эта презентация в стиле старой школы работает весьма неплохо, и персонажи отнюдь не теряются на фоне. Ваша задача — пробираться по горящему зданию, находить охваченных паникой людей и швырять их в окно, что усложняется наличием на вашем пути препятствий; в основном это огонь.

Вам выдан шланг, чтобы гасить огонь и пробивать двери, но если вы опустошите запасы воды, придется подождать

«Находить охваченных паникой людей и швырять их в окно.»



► **Вот что происходит, если застояться на месте: все будет обжато пламенем.**

несколько секунд, пока они воспламятся. Также есть тепломер, который расширяется, когда вы проводите время у огня.

Игра хороша благодаря очень подвижному управлению. Немало платформенных игр имеют отличную проработку уровней, но портят впечатление слишком медленным откликом. А вот *Mr Rescue* реагирует мгновенно, и если вы сыграли в ящик, то сами виноваты.

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Дистрибутивы

В данный момент на фронте дистрибутивов тишина. Ее не объяснить летними каникулами: тогда мы бы увидели кучу дистрибутивов из Австралии. Это может быть затишье перед бурей, так как часть основных дистрибутивов находится на стадии бета-тестирования. А может быть, «эффект Ubuntu» — столько дистрибутивов ныне основано на Ubuntu, что где-то через месяц после выхода официального релиза Ubuntu можно ожидать сущего шквала релизов. Или же все слишком заняты переоценкой своих опций шифрования и безопасности, чтобы заниматься программированием (спасибо, АНБ).

Сейчас отличное время для выпуска нового дистрибутива: на него скорее обратят внимание, чем в конце апреля или октября, когда начинается всеобщая шумиха. Эта ситуация осложняет задачу поиска чего-то достойного для DVD, но и делает эту задачу куда интереснее. Некоторые из дистрибутивов, которые мы рассматривали при составлении DVD, были не вполне оптимальны, но многие являли собой реальную альтернативу стандартным крупным игрокам, которые нередко и сами начинали как альтернативные дистрибутивы или ответвления. Приятное напоминание, насколько творческим является сообщество.

Neil

» Важно ВНИМАНИЕ!

Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

Дистрибутив Linux

LXLE



Э то еще один дистрибутив, способный хорошо работать на старом оборудовании, однако это вовсе не означает, что владельцы более новых и более мощных компьютеров должны его игнорировать. Работа с чем-то вроде используемого здесь LXDE, после того, как вы привыкли к Gnome, Unity или KDE — это истинное откровение, настолько

он отзывчив и плавен в работе. Некоторые легкие системы бывают слишком утилитарными, но LXLE и выглядит тоже достойно. Дистрибутив основан на релизе Lubuntu с долгосрочной поддержкой, поэтому не ожидайте найти самые современные программы — рассчитывайте на быструю, ровную и стабильную систему.

Дистрибутив Linux

OS/4 OpenLinux

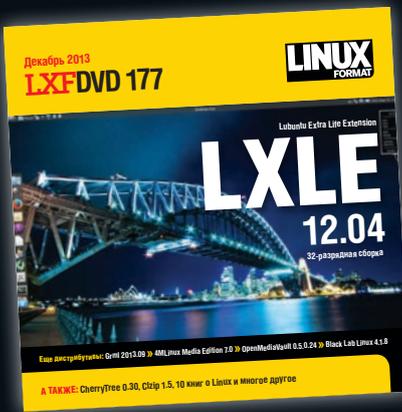
С лишком часто дистрибутивы, в частности, настольные, сравнивают с Windows, однако не все начинают с этой операционной системы. Многие из нас в свое время попробовали на зуб Amiga (а некоторые уходят корнями еще глубже, но это уже другая история), и они, возможно, узрят нечто знакомое в OS/4 OpenLinux — и не в последнюю очередь в его логотипе. Фактически, он настолько похож на AmigaOS... аж на сайте пришлось указать, что это не филиал Amiga Inc! Рабочий стол здесь — KDE, и это — 64-битная версия, поэтому не пытайтесь запустить ее на 7-МГц CPU 68000.

Grml

Не всегда все идет по плану, поэтому очень неплохо предусматривать «запасной выход». Для нас это

означает наличие под рукой дистрибутива live, желательно такого, в котором есть достаточно инструментов на случай экстренной ситуации. Хорошим примером такого дистрибутива является Grml, который мы включили в 32- и 64-битных версиях.





Новичок в Linux? Начните отсюда!

- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент MS Office?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

Ответы приводятся в [Index.html](#) на диске.

Из журнала

Open Media Vault

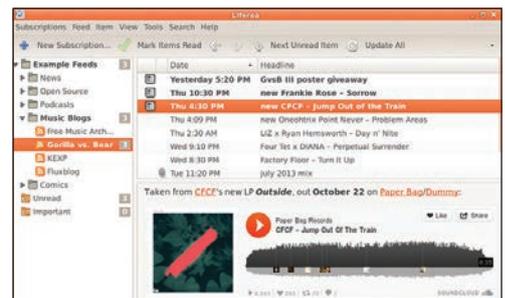
Следуйте нашему руководству с устанавливаемым ISO-образом.

Сравнение

Большинство читалок RSS, которые мы сравнивали в этом месяце.

OS/4 OpenLinux 13.7

Мы рассказывали о нем в прошлом месяце — теперь пробуйте сами.



LXFHotPicks

CherryTree

Приложение для иерархических заметок с функциями rich text и выделения синтаксиса.

ClassicMenu

Классическое меню приложений в стиле Gnome для тех, кто предпочтет его меню Unity Dash.

Clzip

Clzip — программа архивации данных без потерь, на алгоритме LZMA.

4MLinux Media Edition

Ориентированный на мультимедиа мини-дистрибутив на базе стандартного 4MLinux.

Виджет Equake

Монитор землетрясений, реализованный как апплет Gnome и плагин панели Xfce.

Jpgind

Генерируйте индивидуальные статические веб-галереи из файлов изображений JPEG и RAW.

MrRescue

2D-экшн в стиле аркады, суть которого в том, чтобы эвакуировать людей из горящих зданий.

Lightspark

Flash-плеер и плагин браузера с открытым кодом.

SavageWheels

2D-аркада с миссией крушить машины; участвует до 4-х машин.

Lynx

Чисто текстовый веб-браузер, который работает в терминале.

Pragha

Быстрый, легкий, да еще и полный плеер музыки, на GTK и SQLite.



И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязательные присутствовать в каждой операционной системе.

Hardinfo Инструмент для тестирования системы.

Plop Простой менеджер загрузки для запуска операционных систем.

RaWrite Создавайте загрузочные диски в Windows.

SBM Независимый от ОС менеджер загрузки с простым в использовании интерфейсом.

WvDial Соединяйтесь с Интернетом через телефонный модем.

Чтение

Книжная полка

The Cathedral and the Bazaar [Собор и Базар] Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

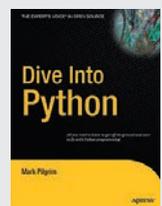
Linux Kernel in a Nutshell [Ядро Linux в двух словах] Введение в ядро, написанное магистром-хакером Греггом Кроа-Хартманом [Greg Kroah-Hartman].

Справочник администратора Debian Руководство для системных администраторов.

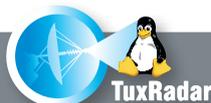
Словарь Linux Linux от А до Z.

Окунитесь в Python Мастер-класс по этому популярному языку.

Руководство по созданию скриптов Bourne Shell Начните осваивать скрипты оболочки.



Подкасты



S5 E16 ROT-13

Новости [News] Шифрование умерло, Новый Humble Indie Bundle, FreeBSD бросает GCC, Побочный продукт Intel и Ubuntu.
Открытое голосование [Open Ballot] Лучший дистрибутив новичкам
Открытия недели [Discoveries of the week] Manjaro очень хорош, CyanogenMod, Ampache и RasbyFi.

S5 E15 Bomba Kabisa

Новости [News] Groklaw закрывается, Новая Зеландия запрещает софтверные патенты, распродажа телефонов Firefox.
Открытое голосование [Open Ballot] Где будет СПО через 22 года?
Открытия недели [Discoveries of the week] Android Photo Sphere, Python — лучший, руководство по VirtualBox длинновато.



Пропустили номер?



Закажите его через «Линуксцентр» по адресу www.linuxcenter.ru! Журналы доставляются и в печатной, и в электронной форме, так что с момента открытия браузера до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:

LXF174

Сентябрь 2013

- » **Пышное застолье** Рабочие столы на все вкусы
- » **Храним анонимность** Дистрибутивы вас прикроют
- » **OpenColorIO** Долой разнобой в колорите
- » **Кино на разные лады** Три учебника по видеоредакторам

LXFDVD: Mint 15 (сборка от LXF с рабочими столами Mate, Cinnamon, KDE и Unity), Bodhi Linux, Siduction, Snowlinux, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_174/
PDF-версия: shop.linuxformat.ru/elxf_174/



250 руб.

LXF175

Октябрь 2013

- » **Сисадмины будущего** Прокладываем новые пути
- » **Звуки музыки** Доверьтесь Поток
- » **putp.io** Движок для открытых социальных сетей
- » **Сайт в гуще жизни** Как раскрутиться в Интернете

LXFDVD: Fedora 19 (64-разрядная сборка с Gnome 3), Arch Linux, Knoppix, Tails, Overclockix, CentOS, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_175/
PDF-версия: shop.linuxformat.ru/elxf_175/



250 руб.

LXF176

Ноябрь 2013

- » **Строим свой дистрибутив** Даже со своим логотипом
- » **Скупым рыцарям** Коллекция под контролем
- » **Open vSwitch** Виртуальным серверам — единоначалие
- » **Я тучка, тучка, тучка...** Облака нынче в моде

LXFDVD: ROSA Fresh R1 (64-разрядная сборка с KDE), Peppermint 4, Sonar 13.10, Ubuntu 13.10, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_176/
PDF-версия: shop.linuxformat.ru/elxf_176/



250 руб.

А чтобы не упустить ничего, оформите подписку! Все, кто подписался на печатную версию журнала через www.linuxformat.ru/subscribe или www.linuxcenter.ru, получают электронную версию в подарок! Через shop.linuxformat.ru вы также сможете приобрести предыдущие выпуски LXF (если успеете).

Подписывайтесь на www.linuxformat.ru/subscribe/

Телефоны отдела подписки:

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



Linux Format ВКонтакте

Вступайте
в нашу группу
vk.com/linuxform

На странице
журнала Linux Format
ВКонтакте вы найдете:

- » Множество новостей о Linux
- » Статьи из архива LXF
- » Анонс свежего выпуска LXF и часть статей из него
- » Живое общение и онлайн-консультацию по подписке на наше издание

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя_программы-1.0.1.tgz** — тот же файл, что и выше этажом по списку: «tgz» — это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

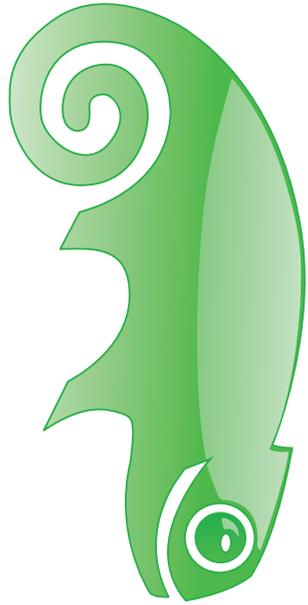
Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: dists@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

А ТАКЖЕ: Yast 3.0, Xen 4.3, KVM/DEMU 1.6 и другие обновления

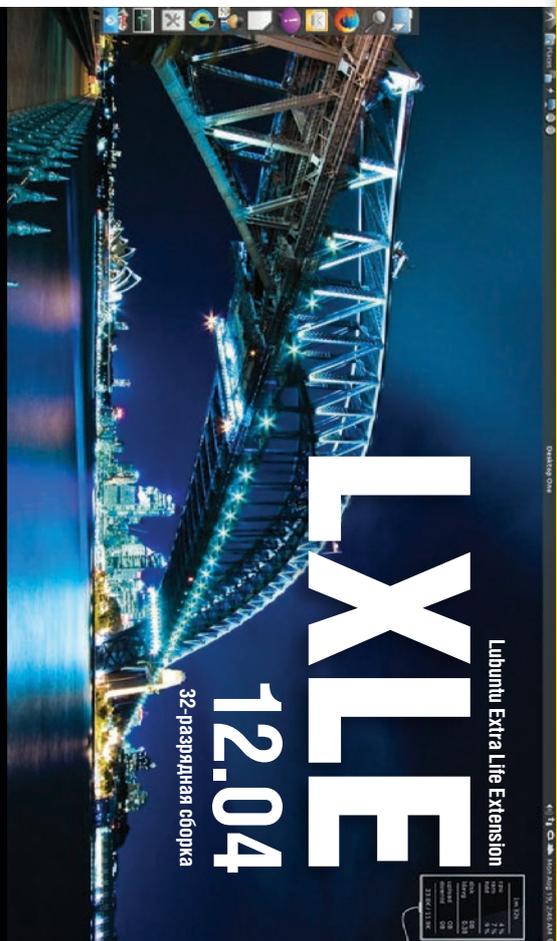
openSUSE 13.1



Ядро 3.11 » Gnome Shell 3.10 » KDE Plasma 4.11 » Firefox 24 и Chromium 21
64-разрядная сборка

А ТАКЖЕ: CentOS 6.30, Sclib 1.5, 10 книг о Linux и многое другое

Еще дистрибутивы: Sml 5.013.09 » Dell Linux Media Edition 7.0 » OpenMediaVault 0.5.0.24 » BackLan Linux 4.1.8



Linux Extra Life Extension

32-разрядная сборка
LXLE
12.04

Декабрь 2013
LXF DVD 177

LINUX
FORMAT

Декабрь 2013
LXF DVD 177

LINUX
FORMAT

Содержание

LINUX FORMAT

Сторона 1

ДИСТРИБУТИВЫ

Gnpl 2013.09 LiveCD на базе Debian с набором программ для системных администраторов, 32- и 64-разрядные сборки (ISO-образы)
LXLE 12.04 Версия LXubuntu на основе LTS-релизов, 32-разрядная сборка (загрузка с LXFDVD)
Black Lab Linux 4.1.8 (ранее именовался OS/4 OpenLinux) Дистрибутив с рабочим столом в стиле Amiga, 64-разрядная сборка (ISO-образ)

Сторона 2

ДИСТРИБУТИВЫ

openSUSE 13.1 Обновленная версия популярного дистрибутива. Установочный DVD, 64-разрядная сборка (загрузка с LXFDVD)

ДОКУМЕНТАЦИЯ — 10 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

Bash Scripting Подробное руководство по программированию на Bash
Bourne Shell Scripting Начальное руководство по программированию на Bash
Cathedral Bazaar Классический текст Эрика Раймонда [Eric S Raymond] «Сбор и базар»
The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
Dive Into Python Учебник по программированию на Python
Intro to Linux Начальное руководство по Linux

Linux Dictionary Словарь Linux, объясняющий специфическую терминологию
Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Греггом Кроа-Хартманом [Greg Kroah-Hartman]
System Administrators Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

НОТРИКСЫ

4MLinux Media Edition 7.0 Мини-дистрибутив с уклоном в мультимедиа
CherryTree 0.30.1 Приложение для заметок
ClassicalMenu 0.08 Традиционное меню приложений для Unity
Czip 1.5-rc1 Инструмент архивирования

EqnakeWidget 1.3.2 Мониторинг землетрясений
Jrjrd 0.87 Генератор web-галереи
LibRescue 1.02 Платформенная игра 2D в стиле ретро
Lightspark 0.72 Flash-плеер с открытым кодом
Linux 2.8.8-dev-16 Web-браузер, работающий в режиме командной строки
Praha 1.1.21 Музыкальный плеер
Savage Wheels 1.5 Игра, аркада с автомобильными авариями

ПРОГРАММЫ

OpenMediaVault 0.5.0.24 Дистрибутив для создания сетевых хранилищ данных (NAS) на базе Debian
Программы для чтения RSS, рассмотренные в рубрике «Сравнение»

Пожалуйста, перестаньте использовать именем Аджного Диска озвучивать с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех старых платформах, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных в случае 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, скачайте программу для загрузки с нашего сайта: www.linuxformat.ru

Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

Создание установочных дисков при помощи 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/cdcrecord*. 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 *cdcrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прожечь диск у них. Опять-таки, вам подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.



UNIXFORUM.org



Есть вопросы?
Задавайте!

<http://unixforum.org/>

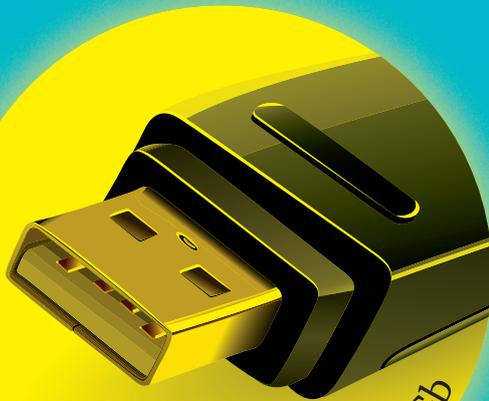
Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

Широкая сеть представительств
в разных городах России
позволит вам оптимизировать
процессы логистики и доставки товара

ПОДРОБНЕЕ О ПАРТНЕРСКОЙ ПРОГРАММЕ:
WWW.LINUXCENTER.RU/PARTNER/



ДИСТРИБУТИВЫ LINUX ТЕПЕРЬ НА USB FLASH



Linux Mint 15



Ubuntu 13.04

А ТАКЖЕ
версии для юриди-
ческих лиц —
с лицензионным
договором
присоединения

USB flash 8 Gb

495 рублей

www.linuxcenter.ru/shop/distros/usb/



UnixEducationCenter



Certified Training Partner

Обучение & Сертификация **RHCSA + RHCE**

Впервые в Санкт-Петербурге:

RHCVA®



RHCDS®

Санкт-Петербург, Черноморский переулок, дом 4

Телефон: +7 (812) 611-15-75

mail@unixedu.ru 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/



KICKSTARTER

В январском номере

Сбудем мечту всем миром

Не дают банки ссуду — и не надо. Рассказываем, как сделать, чтобы ваш проект взлетел, при помощи людей со всего мира.

Внутри Qt

Сегодня — мобильники, Windows, OS X и Linux, а завтра — весь мир! Узнайте, почему за Qt 5 — будущее.

Drupal

В стране Оз не хватает разработчиков Drupal; срочно изучите его и махните туда.

Мастерим рабочий стол

Выберите оконный менеджер, панель задач, виджеты и пр. по своему вкусу и сколотите рабочий стол всем на удивление.

Содержание будущих выпусков может меняться — вдруг без Бена мы увянем в цвету...

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 3000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова, Ирина Шулакова

Редактор диска

Кирилл Степанов

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филипов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

ЗАО «Мезон.Ру»

Отпечатано в ООО «Ланпін»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 8456

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон [Graham Morrison] graham_morrison@futurenet.com

Заместитель редактора Эндрю Грегори [Andrew Gregory]

andrew.gregory@futurenet.com

Научный редактор Бен Эверард [Ben Everard] ben_everard@futurenet.com

Выпускающий редактор Крис Торнетт [Chris Thornett]

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

[Elrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Джоно Вэсон [Jono Vascon], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Бен Эверард [Ben Everard], Марко Фиоретти [Marco Fioretti], Дэмьен Мак-Ферран [Damien McFerran], Грэм Моррисон [Graham Morrison], Лес Паундер [Les Pounder], Майк Сондерс [Mike Saunders], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Евгений Балдин, Артем Зорин, Андрей Прахов, Андрей Ушаков, Алексей Федорчук, Игорь Штомпель, Сергей Яремчук

Художественные ассистенты Ники Готубед [Nicky Gotobed],

Донна-Мари Скразе [Donna-Marie Scrase]

Иллюстрации Шейн Коллиндж [Shane Collinge],

Саймон Миддлвик [Simon Middleweek]

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 01225 442244, email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):

пр. Медиков, д. 5, корп. 7

Тел. +7 (812) 309-0686

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел./факс +7 (499) 271-4954

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания Linux Format, являются собственностью или 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 повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

«GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журнале, издаваемом Future plc group company, обращайтесь на сайт <http://www.futureplc.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

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.

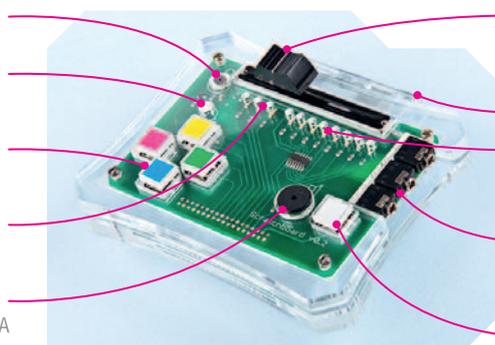
ДАТЧИК ЗВУКА*

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

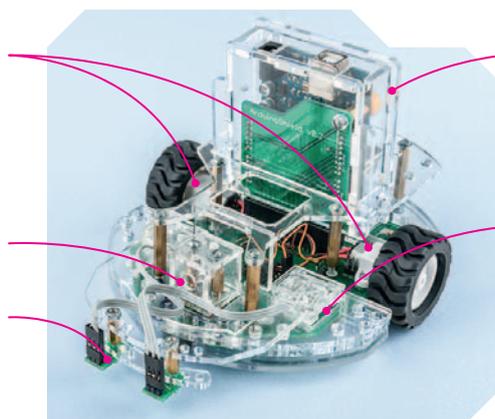
обеспечивает возможность:

- 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)

СВЕЖИЕ РАЗРАБОТКИ!

Новинка!

2 x 2 TB SATA
Enterprise



PX60

2 x 240 GB
SATA SSD
Data Center
Series



PX60-SSD

2 x 4 TB SATA
Enterprise



PX70

2 x 480 GB
SATA SSD
Data Center
Series



PX70-SSD

ВЫДЕЛЕННЫЙ СЕРВЕР PX60

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 2 TB 6 Гбит/с SATA 7200 об/м Software-RAID 1 Enterprise класс
- 20 TB Трафик*

2600

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX60-SSD

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 240 GB 6 Гбит/с SATA SSD Data Center Series
- 20 TB Трафик*

3000

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX70

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 4 TB 6 Гбит/с SATA 7200 об/м Software-RAID 1 Enterprise класс
- 20 TB Трафик*

3000

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX70-SSD

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 480 GB 6 Гбит/с SATA SSD Data Center Series
- 30 TB Трафик*

3700

рублей в месяц + Установка 3700 рублей

ВКЛЮЧЕНО:

- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта



Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выбирайте более чистое будущее вместе с Hetzner Online!

RU.HETZNER.COM



* Нет платы за превышение. При превышении 20 TB/месяц (PX60, PX60-SSD, PX70), 30 TB/месяц (PX70-SSD) скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 80 рублей за каждый дополнительный TB.