

Крутейший* и лучший журнал про Linux в России

* крутизна = (количество слов x ценность)

LINUX FORMAT

Главное в мире Linux

Май 2012 № 5 (157)

Linux — лидер

Забудьте OS X и Windows: ОС лучшего качества вы получите бесплатно и прямо сейчас



Откроем Pandora

» Разработанный сообществом ящик с сюрпризами с. 18



Парни из Wikitotics

“ У раннего Монти Пайтона есть скетч про венгерский разговорник... ”

Они хотят выучить вас вьетнамскому с. 38

Контроль версий
Хостинг проектов
» Ищем лучший репозиторий для кода

Базы данных
Освоим MySQL
» Там, где господствуют полигамные связи

Редактор фото
Эффекты ЛОМО
» Уклон ваших цифровых фотографий в ретро



LibreOffice 3.5

» Достоин ли этот релиз звания лучшего из всех? с. 12

СУПЕР-БЛЕСТЯЩИЙ DVD
Четвертая загрузка Sabayon 8, Chakra, LXF Open SUSE и #10 и более того!

ПЛЮС!
Попрыгунья-стрекоза – альтернатива пингвину с. 58

Также в номере...

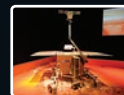
Вовлечем молодежь

» Научите детишек программировать, и светлое будущее обеспечено с. 48



Тукс в космосе

» NASA рассказывает про пингвинов по плечам Ориона с. 52



Arduino

» Организуем взаимосвязь плат – зря, что ли, радио изобрели с. 80



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

Linux center
www.linuxcenter.ru

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

Кто мы

Иногда приятно, подводя итоги, увидеть, как замечательно обстоят дела. И мы спросили нашу команду-ух: что вам больше всего нравится в Linux?



«Пятая свобода»

» В статье «Linux – лидер» Маянк Шарма в очередной раз доказал, что по количеству инноваций и скорости их появления свободная ОС оставила проприетарные аналоги далеко позади. Это наблюдение справедливо не только для Linux, но и для всего мира свободного ПО в целом. И сам собой напрашивается вопрос: а почему? Разве в Microsoft, Oracle или IBM работают менее способные и не столь креативные разработчики? Да вроде бы нет...

Причина, на мой взгляд, проста и напрямую связана с идеологией свободного ПО. Дело в том, что в самом начале проекта, когда есть только идея и немного кода, авторы избавлены от необходимости доказывать его важность многочисленным руководителям департаментов и прочим вице-президентам по развитию. Задача практически неразрешимая – поверьте очевидцу, много лет проработавшему в американской софтовой фирме.

А в случае свободного ПО, если предлагаемый код нужен самим авторам идеи, можно начинать работу. И инновации не умирают при рождении, столкнувшись с корпоративной бюрократией, а доживают как минимум до релиза 0.1. А там уж как получится.

Таким образом, можно говорить о «пятой свободе», в дополнение к тем четырем, что сформулировал великий RMS: свободе творить и, тем самым, делать мир чуть совершеннее.

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

Проблемы с дисками: disks@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

Web-сайт: www.linuxformat.ru

» Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15

» Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112



Гэри Уокер

Linux – это та штука с обгрызанным яблоком, которую придумал какой-то Жобс и все изменил?



Эндрю Грегори

Лучшее – буква X: она напоминает и футуристический вид века космоса, и поцелуй. Почувствуйте любовь X.



Эфраин Эрнандес-Мендоса

Благодаря Linux я интеллигентно выгляжу перед людьми, которые про него не знают.



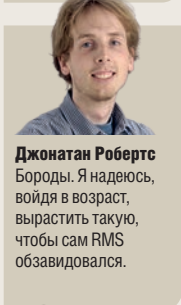
Бен Эверард

Пингвины. Раньше я предпочитал папуанских – Генту, но теперь люблю и императорских.



Маянк Шарма

Конечно, пингвины. Мне нравится, как тает шоколад, когда я макаю их в чай.



Джонатан Робертс

Бороды. Я надеюсь, войдя в возраст, вырастить такую, чтобы сам RMS обзавидовался.



Майк Сондерс

Название Linux мне претит. Ну кто столь суетен, чтобы назвать ОС в честь себя? Невежа и грубиян, вот кто.



Валентин Синецын

Его можно чинить – коды-то доступны. А можно – не чинить, и он все равно будет работать.



Ник Вейч

Лучшее в Linux – он наделяет меня властью над армией роботов на Arduino в моем сарае.



Сюзан Линтон

В Lintopix мне нравится, что я могу нарушить законы природы, установив Gnome и KDE одновременно на одной машине.



Шашанк Шарма

Не слишком ли будет в лоб – сказать, что он бесплатен, в нем нет вирусов и, видя вертящийся куб, я так и лизнул бы экран?



Нейл Боткин

Как элитный пользователь Gentoo, я знаю, что каждый байт в нем компилировал я, и это фантастика по определению.

Содержание

Согреты теплой волной гордости за использование свободного ПО...

Обзоры

LibreOffice 3.5 12

Флагман офисных пакетов свободного ПО за последние месяцы капитально улучшился. Разве нет?

Explay Informer 701 14

Малобюджетный и тем не менее полнофункциональный планшет на Android без особого труда превращается в Linux-машину.

Cinnamon 1.2 16

Не успели мы притерпеться к Gnome 3, как появилось это отличное ответвление. Приятно снова ощутить себя в привычной обстановке.

VLC 2 17

Это приложение для воспроизведения медиа вызывает, как марочное вино, Терри Воган или Джордж Клуни.



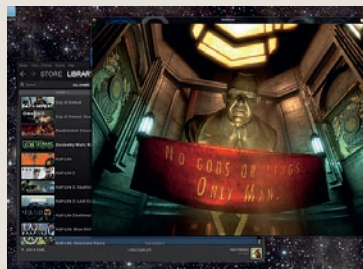
➤ Будь мы медиа-файлами, мы бы сдались проигрывать только VLC.

Pandora 18

В ситуации, когда все зло мира вырывается на волю, людям остается только надежда. И игровая консоль.

CrossOver 11 19

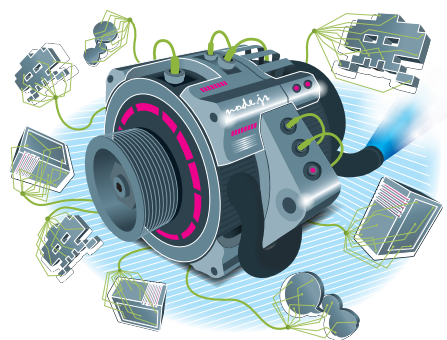
Тем, кому нужны Windows-приложения без загрузки в Windows, это понравится. Игры и офис укупорены в один флакон.



➤ Ба, да это Steam — и тем не менее работает в Linux!



Сравнение: Хостинг проектов **с. 32**



Что за штука — Node.js? **с. 62**

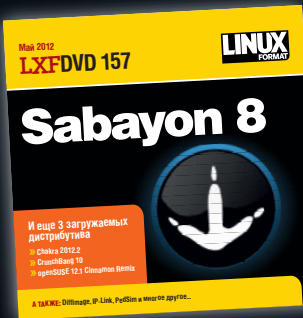
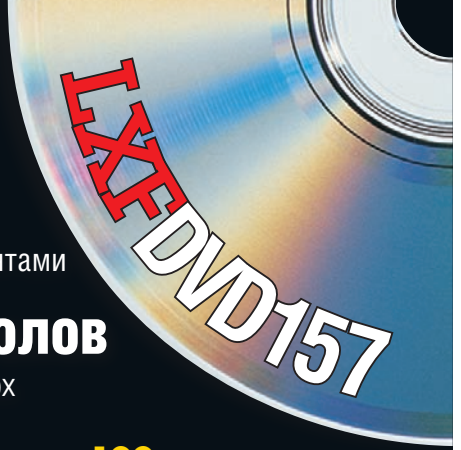
Люди говорят



Учителя — люди занятые, им некогда подучить CSS и HTML и набросать сайтик

Парни из Wikiotics — о наблевшем очевидном **с. 38**

На вашем бесплатном DVD



Sabayon 8

» Потомок Gentoo с классными инструментами

Изобилие рабочих столов

» Попробуйте Cinnamon, KDE 4.8 и OpenBox

ПЛЮС: Горячие новинки и коды к учебникам... **с. 102**

Ищите в этом номере



Вовлекаем молодежь 48

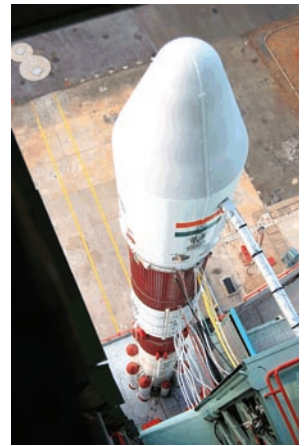
Пусть изучают полезные вещи.

Linux: космическая одиссея 52

Придется вам поверить, что пингвины летают.

Сисадминам 64

А это у нас что? Разберитесь сегодня!



Пропустили номер?

Узнайте на с. 104, как получить его прямо сейчас!



Учебники

Фотография
Эффекты ЛОМО 70
Наш эксперт расплывается по краям, теряет фокус и видит радужные глюки.

Кодирование видео
Handbrake/FFmpeg 74
Полки ломятся от коробок с DVD; пора перекидывать киношки либо в Сеть, либо на жесткий диск.

Безопасность
Удаленный вход 78
Получите доступ к вашей машине отовсюду, как будто вы уселись перед ней – однако не оставляя злодеям шансов на вторжение.

Электроника
Arduino 80
И плата с платой говорит... да не как-нибудь, а при посредстве радиоволн.

Языки программирования
Erlang 84
То, что закодировано, должно подаваться и декодированию. Это касается также и битовых строк.

Для продвинутых
SQL 88
Реляционные базы данных станут вашими друзьями. Правда, не лишенными некоторых странностей.

Постоянные рубрики

Новости 4
GNU/Линуксцентр скорчился с Red Hat, которая становится миллиардером, у скелетов все под контролем, а роботы нанимаются в архивариусы.

Android 22
На повестке дня – священные права root, борьба с тивоизацией и механизмы взаимодействия с Linux.

Сравнение 32
Куда пристроить ваш код-шедевр, когда вы его создадите.

Интервью LXF 38
Попробем изучать иностранные языки по методике **Нейла Салливана** и **Лорана Савайета**.

Что за штука 62
Node.js упрощает сборку приложений, запускаемых как сервер.

Рубрика сисадмина 64
Что было бы, если бы команды UNIX изобретали франкофоны?

Ответы 92
ПРОБЛЕМЫ LINUX РЕШЕНЫ!
Малоохотная мышь, взбесившийся VirtualBox – и другие проблемы, все решено!

Hotpicks 96
Отведаете горяченького: лучшие в мире новинки свободного ПО.

Диск Linux Format 102
Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 104
Еще не поздно заказать любой из предыдущих выпусков журнала.

Школа LXF 108
Заставим виртуального робота сновать по лабиринту, обвешавшись датчиками.

Через месяц 112
Тот факт, что вы параноик, не гарантирует отсутствия слежки за вами.



ГЛАВНОЕ GNU/Линуксцентр — партнер миллиардера » ROSS 2012 » Слежка за скелетом! » Робот-архиватор » Free Software Awards 2011 » Торвальдс в суде

RED HAT УКРЕПЛЯЕТ ПОЗИЦИИ В РОССИИ

GNU/Линуксцентр как премьер

Компания повысила уровень партнерства с Red Hat, став Premier Business Partner.

16 марта 2012 г. компания GNU/Линуксцентр – авторизованный партнер компании Red Hat – объявила о получении высшего статуса Red Hat Premier Business Partner, подтверждающего профессиональный уровень в области поставок программного обеспечения Red Hat, проектной деятельности и технической поддержки.

Red Hat – лидер в разработке программного обеспечения с открытым кодом. Компания занимается разработкой, поставкой и поддержкой решений на базе GNU/Linux и свободного программного обеспечения с 1993 года и, таким образом, является старейшей крупной организацией, работающей в этой области.

Компания Red Hat предъявляет высокие требования к качеству своих продуктов и к работе партнеров – их репутации и опыту. Получение GNU/Линуксцентром статуса Red Hat Premier Business Partner стало возможно в результате выполнения компанией совокупности условий по наличию в штате не менее четырех инженеров, прошедших сертификацию RHCE (Red Hat Certified Engineer), сертифицирован-

ных специалистов по продаже продуктов (Red Hat Certified Sales Person), а также высокому уровню продаж в корпоративном секторе и признанию клиентами качества предоставляемых услуг.

В настоящий момент подобного статуса Red Hat в России удостоились только три компании, включая GNU/Линуксцентр.

» Только лучшие получают высокий статус Red Hat Premier Business Partner.

«Подобного статуса в России удостоились только три компании.»

При этом в Северо-западном Федеральном Округе GNU/Линуксцентр – единственный партнер Red Hat уровня Premier Business Partner.

«Мы сотрудничаем с компанией Red Hat и ее мастер-дистрибьютором на протяжении долгого времени. Наши достижения – большое количество выполненных проектов, высокая квалификация специалистов – в этом году подтверждены получением статуса Red Hat Premier Business Partner. За последнее время ре-



» Рубрику готовил
АРТЕМ ЗОРИН



redhat.

PREMIER
BUSINESS
PARTNER

шения компании Red Hat пользуются все большим спросом, который подтвержден более чем двукратным ростом оборотов продаж продуктов вендора в 2011 году по сравнению с предыдущим периодом. Мы и дальше планируем активно развивать направления Red Hat, сохраняя динамику роста. Новый статус – свидетельство того, что клиенты, обращаясь в нашу компанию, всегда получают качественные услуги по поставке, внедрению и дальнейшему обслуживанию продуктов Red Hat», отмечает Сергей Владимирович Логутов, ответственный за развитие направления Red Hat коммерческий директор компании GNU/Линуксцентр.

Видеозапись совместного семинара GNU/Линуксцентр и Red Hat можно посмотреть на сайте Lektorium TV: <http://www.lectorium.tv/course/?id=22859>.

GNU/Линуксцентр — это мы

Компания GNU/Линуксцентр – лидер российского рынка дистрибуции операционных систем GNU/Linux и другого свободного программного обеспечения. Главная задача GNU/Линуксцентра – популяризация и продвижение свободного ПО в России. Компания GNU/Линуксцентр была основана в 2000 году как дистрибьютор и издатель свободного программного обеспечения, с собственным интернет-магазином, новост-

ным каналом, интернет-библиотекой по СПО. Сейчас GNU/Линуксцентр занимается предоставлением услуг по внедрению и поддержке СПО, спонсирует свободные проекты и издает журнал Linux Format – единственный журнал в России, полностью посвященный свободному программному обеспечению. Офисы GNU/Линуксцентра расположены в Москве, Санкт-Петербурге, Нижнем Новгороде, Владивостоке, Екатеринбурге.

softline®

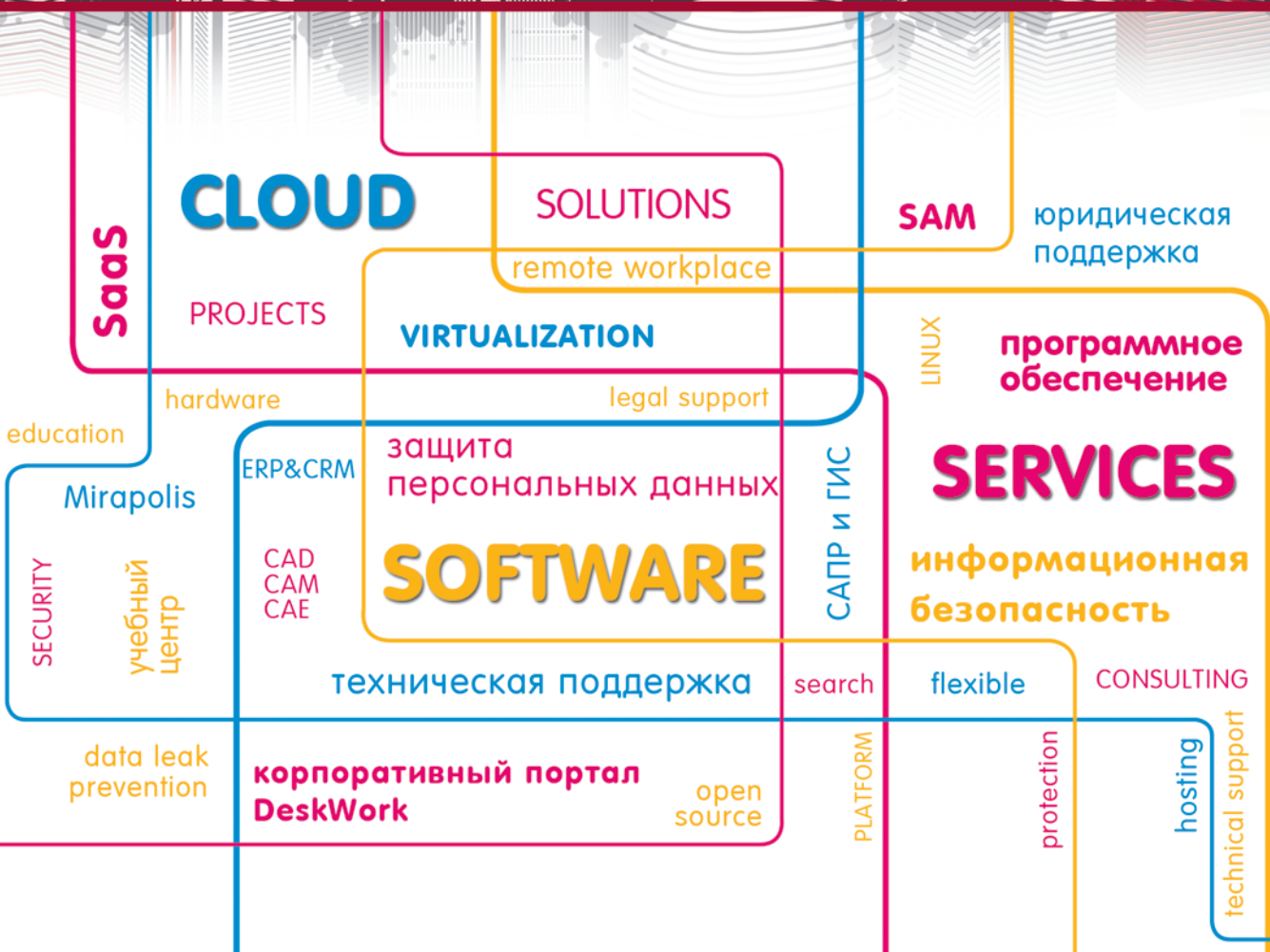


Services

Software

Cloud

ИТ-архитектура вашего бизнеса



КОНФЕРЕНЦИИ ОБЪЕДИНЯЮТ!

Москва стала столицей СПО



ROSS'2012 сдружила разработчиков и пользователей СПО-решений.

12 апреля 2012 года в московском отеле «Ренессанс» (Олимпийский проспект, 18/1) состоялся Russian Open Source Summit 2012, посвященный развитию СПО в России. Организаторы саммита – Ассоциация разработчиков СПО (РАСПО), ведущий IT-еженедельник «PC Week/RE» и «Академия информационных систем».

Конференция прошла при участии НИУ «Высшая школа экономики», Федеральной службы судебных приставов, МГУ им. Ломоносова, ООО «Центр безопасности информации», «ГНУ/Линуксцентр», других организаций и компаний.

Ключевыми вопросами, рассмотренными на саммите ROSS'2012, стали: реализация плана по переходу государственных органов власти и бюджетных учреждений на СПО, создание Национальной программной платформы (НПП), практическое использование СПО в различных отраслях экономики и государственном управлении, разработка продуктов на базе свободного ПО для малого и среднего бизнеса, сертифицированные и защищенные программные решения на базе открытого кода, обучение работе с открытым ПО.

На сегодняшний день партнерами саммита являются ассоциация АП КИТ, НП

РУССОФТ, а также Институт проблем информатики РАН. Генеральным спонсором выступает компания Red Hat, деловым партнером – корпорация Microsoft. Спонсорами саммита стали российские компании-разработчики «НЦПР», «Пингвин Софт» и корпорация EMC.

В рамках пленарного заседания и в тематических секциях выступили около 50 докладчиков – представителей государственных ведомств и коммерческих организаций, науки и образования, разработки, а также заказчики свободных решений из различных отраслей экономики. Отдельные секции были посвящены созданию защищенных и сертифицированных программных продуктов на базе открытого кода, СПО для облачных сервисов и мобильных платформ, другим актуальным направлениям развития информационных технологий. Заключился форум совместным круглым столом РАСПО и IT-еженедельника «PC Week/RE», под названием «Национальная программная платформа – точка роста отечественной IT-индустрии».

Есть изменения

После прошлогоднего саммита ROSS'2011 отечественная отрасль СПО шагнула далеко вперед. Реализованы крупные проекты по внедрению СПО в государственном и коммерческом секторе, а также разработаны и приняты прототипы базовых программно-технических

компонент НПП. Вопросы создания Национальной программной платформы сегодня очень интересуют сообщество российских разработчиков и пользователей СПО, поэтому данная тема стала одной из главных в повестке дня прошедшего мероприятия.

ОТКРЫТОЕ МАНИПУЛИРОВАНИЕ

Skeltrack – аналог Microsoft Kinect

Первый открытый проект для отслеживания перемещения скелета человека.

Аннотирован проект Skeltrack, развивающий свободную библиотеку для отслеживания в режиме реального времени движений скелета человека перед камерой с датчиком глубины. Библиотека позволяет в наборе меняющихся изображений выделить и отследить перемещение контрольных точек, сопоставленных с конечностями и головой человека, давая возможность распознавать естественные движения и использовать их в качестве управляющих жестов или отождествлять с движениями персонажа игрового приложения.

Skeltrack применяет набор математических алгоритмов и эвристических методов для реконструкции человеческого скелета, не требует калибровки положения человека и не использует дополнительных баз данных с описанием возможных поз. Пока библиотека поддерживает отслеживание только одного человека и может контро-

лировать движение 7 контрольных точек (голова, плечи, локти, кисти рук). Оценить возможности Skeltrack можно на сайте <https://vimeo.com/38875885>.

Библиотека не зависит от типа устройств, но наиболее доступное устройство с датчиком глубины – Microsoft Kinect, для которого Skeltrack может сочетаться с открытой библиотекой *GFreenct*, GObject-обвязкой для открытой кроссплатформенной библиотеки *libfreenect*, развиваемой сообществом OpenKinect и позволяющей организовать прием видеопотока с данными глубины от камеры Kinect.

Код библиотеки Skeltrack распространяется под лицензией LGPLv3. Это первый открытый проект, решающий задачу отслеживания скелета для Kinect. До этого предлагалось использовать либо проприетарный пакет Microsoft Kinect SDK, либо полуоткрытый фреймворк OpenNI. Проблема OpenNI состоит в том, что пол-

ностью открытым является лишь базовый каркас, а все дополнительные компоненты, непосредственно реализующие необходимую функциональность, поставляются под лицензией, не допускающей модификацию кода и его использование в сторонних продуктах.

Что такое OpenNI

Открытый проект OpenNI, под лицензиями GPLv3/LGPLv3, создан компанией PrimeSense (Израиль) в 2010 г. Их технологии выбрали устройство Microsoft Kinect. Также был открыт пакет драйверов и сопутствующих компонентов для работы с сенсорами. Цель проекта – обеспечение совместимости и переносимости приложений, промежуточного ПО и устройств на базе естественного взаимодействия с пользователем. Так, OpenNI предоставляет универсальный API для управления приложениями через движения пользователя и голосовые команды. Это не только игры, но и интуитивное управление бытовой электроникой через язык жестов (скажем, переключение канала телевизора).

ПОРА ПРИНЯТЬ УЧАСТИЕ

Конкурс «Лучший свободный проект России в госсекторе 2011»

Сергей Голубев информирует о предстоящем мероприятии, весьма важном для развития свободного ПО.

С момента выхода распоряжения Правительства РФ «О плане перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения (2011–2015 годы)» прошло уже семнадцать месяцев. Срок немалый.

Что удалось сделать за это время? Какие государственные структуры лидируют в этом процессе, а кого можно записать в аутсайдеры? Каковы масштабы внедрения СПО на сегодняшний день? Есть ли какие-либо заметные успехи в данном направлении?

Возможно, ответы на некоторые вопросы поможет получить конкурс «Лучший свободный проект России в госсекторе 2011», проводимый Российской ассоциацией свободного программного обеспечения совместно с журналом *LinuxFormat*. Итоги конкурса будут подведены 10 мая, так что ждать осталось совсем недолго.

Проведение подобного конкурса само по себе явление для нашего общества если не уникальное, то довольно редкое. Согласитесь, не очень часто это самое общество оценивает работу государства.

Поэтому есть все основания надеяться, что итоги конкурса будут максимально объективными. В состав жюри входят представители практически всех ведущих отечественных компаний, занимающихся разработкой и внедрением СПО, а также представители СПО-сообщества, непредвзятость которых не вызывает никаких сомнений.

Несколько смущают сроки проведения конкурса – на все чуть больше месяца. Впрочем, история выполнения госпроекта по разработке прототипа НПП показала, что участники РАСПО умеют мобилизоваться и выполнить проект в практически нереальные сроки. Или это просто тоска по студенческой юности, когда за ночь надо проштудировать то, что лектор излагал в течение полугода? Как бы то ни было, работу в экстремальном режиме у нас никто не запрещал, лишь бы результат был.

Под результатом, разумеется, подразумевается не столько «раздача слонов», сколько тиражирование полезного опыта.

Вдруг какой-то орган государственной власти уже полностью перешел на СПО, а другие про это не знают, продолжая твердить мантры о несовместимости офисных форматов и необходимости потратить весь годовой бюджет страны на обучение служащих премудростям работы в этой непонятной ОС Linux!

Действительно, согласно Положению о конкурсе «Лучший свободный проект России в госсекторе 2011», главными задачами конкурса, в частности, являются:

- » Представление достижений российских органов власти в части внедрения свободного программного обеспечения для создания единой базы лучших практик внедрения СПО на государственном уровне и формирования пула типовых решений.
- » Обмен опытом между органами власти разных уровней в части внедрения СПО решений.

«Есть все основания надеяться, что итоги будут объективными.»

Внедрения могут быть номинированы на участие в конкурсе не только заказчиком или исполнителем проекта, но также и членами оргкомитета или жюри. Если учесть сжатые сроки и традиционную инертность государственных структур, такое решение представляется разумным.

Наконец, самое главное – про объективность оценок членами жюри. Очевидно, без определенной доли субъективизма тут не обойтись. Критерии значимости проекта вряд ли можно строго формализовать, поэтому большую роль будут играть личные мнения экспертов. Как уже написано выше, частично компенсировать эту особенность можно за счет разнообразия жюри. Однако организаторы конкурса решили, что этого будет недостаточно, и указали в положении совершенно определенные критерии, по которым должны оцениваться внедрения. Данный подход позволит минимизировать субъективные предпочтения членов жюри.



- Итак, внедрения будут оцениваться по:
- » стратегической важности проекта развития индустрии;
 - » использованию отечественных решений;
 - » тиражируемости проекта;
 - » масштабу проекта (объему финансирования, количеству исполнителей, заказчиков и пользователей);
 - » роли проекта для популяризации СПО;
 - » сложности проекта;
 - » кросс-платформенности решений.

При таких жестких условиях оценки членам жюри будет непросто отдать предпочтение своим любимчикам, даже если такие у кого-то имеются.

И нельзя забывать по номинацию «Выбор сообщества». В ней победитель будет выбран после публичных обсуждений на профильных форумах и в социальных сетях. Каждый участник сообщества сможет высказать свое мнение, которое обязательно должен учесть оргкомитет.

Подробности см. на сайте www.raspo.ru. Об оценке действий правительства, как это делается у британских коллег, см. на с. 48.

Конкурсные номинации

Итоги конкурса будут подводиться по следующим номинациям:

- » Оборона и безопасность
- » Образование
- » Здравоохранение
- » Исполнительная власть
- » Законодательная власть
- » Судебная власть
- » Региональное управление
- » Выбор сообщества
- » Содействие развитию СПО в государственных органах.
- » Подготовка кадров в области СПО.
- » Облачные инфраструктуры.
- » Портальные решения.
- » Электронные государственные услуги.
- » Документооборот.
- » Внедрение СПО на рабочих станциях.
- » Внедрение СПО на серверах.
- » Специализированное решение на базе СПО.
- » Переход на открытые форматы.
- » Безопасность.

ОТКРЫТЫЙ РОБОТ-СКАНЕР

Gado, друг мелких библиотек

Проект для оцифровки документов на базе платы Arduino.

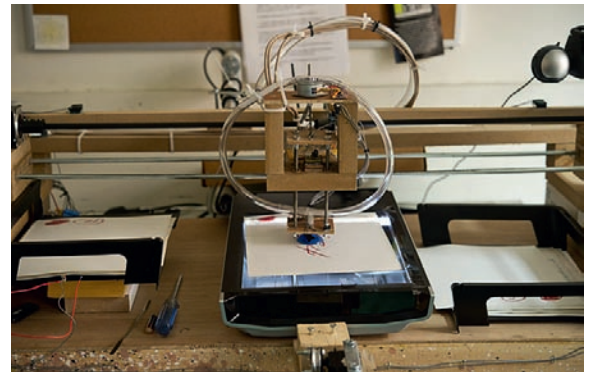
В рамках проекта Gado предпринята попытка создания максимально открытого автономного архивного робота для сканирования и оцифровки книг, изображений и документов. Главная практическая цель, которую преследуют разработчики – это дать небольшим архивам, музеям и библиотекам возможность без покупки дорогостоящих систем автоматического сканирования, своими силами провести оцифровку материалов, подарив вторую цифровую жизнь многим драгоценным документам и книгам уходящей эпохи. В настоящее время прототип открытого робота-сканера Gado уже задействован в проекте оцифровки изображений газеты “Afro American”.

Проект Gado основан в 2010 году и предполагает поэтапное развитие. В настоящее время завершена первая фаза проекта, в рамках которой был создан работающий прототип сканирующей машины. Сейчас в самом разгаре вторая стадия этого проекта – завершается создание Gado 2, отличиями которого являются более высокая скорость сканирования, компактные раз-

меры (в два раза меньше первого прототипа), общая простота эксплуатации (для работы не требуется никаких специальных навыков). Итоговая стоимость сканирующего устройства, которое предлагается собрать из широко распространенных недорогих компонентов, не должна превышать сумму в 500 долларов за экземпляр.

Аппаратная часть устройства базируется на открытом микроконтроллере Arduino. Управление производится с ПК на базе Ubuntu Linux. Все спецификации и схемы открыты и доступны для доработки и улучшения другими энтузиастами. В качестве движка для распознавания текста используется *Tesseract OCR*, для хранения метаданных и каталогизации введенных данных применяется *MySQL*. Для реализации логики работы всей системы на прикладном уровне используется набор скриптов на языке Python, для оцифровки фотографий задействован пакет *gPhoto*, а для управления самим процессом сканирования – SANE.

Лидер проекта Томас Смит [Thomas Smith] подчеркивает, что использование



» «Профессиональные» роботы-сканеры стоят десятки тысяч долларов, в отличие от Gado, который может себе позволить обычная библиотека.

открытых инструментов не только позволило удержать итоговую цену робота в пределах 500 долларов, что было очень важно для проекта, но также предоставило нужный уровень гибкости в доработке и адаптации каждой отдельной его составляющей при решении достаточно необычной задачи. Предварительный заказ сканера Gado 2 уже сейчас открыт через сайт Kickstarter. Финальная версия будет готова к поставкам ориентировочно в августе этого года.

ЛУЧШИЕ ПРОЕКТЫ 2011 ГОДА

Слонов раздает Столлмен

Фонд СПО объявил лауреатов премии Free Software Awards 2011.

Ричард Столлмен [Richard Stallman, он же RMS] официально объявил на конференции LibrePlanet 2012 лауреатов ежегодной премии “Free Software Awards 2011”, учрежденной Фондом свободного ПО (FSF) и присуждаемой людям, которые внесли наиболее значительный вклад в развитие свободного ПО, а также социально значимым свободным проектам.

В первой номинации, вручаемой за продвижение и развитие свободного ПО, премия была присуждена Юкихино Мацумото [Yukihiko Matsumoto], автору языка программирования Ruby. Он уже на протяжении 20 лет участвует в развитии проектов GNU, Ruby и других открытых проектов. В своей ответной речи Юкихино рассказал о своем первом опыте работы со свободным ПО и подчеркнул влияние GNU Emacs на Ruby. Во второй номинации, вручае-



мой проектам, принесшим значительную пользу обществу и способствовавшим решению важных социальных задач, награда присуждена проекту GNU Health, в рамках которого развивается свободная информационная система для автоматизации больниц и медицинских учреждений. Про-

» Отец-основатель движения СПО и Юкихино Мацумото.

ект развивается при участии профессиональных работников здравоохранения со всего мира, и ставит своей целью улучшение жизни неимущих и подъем здравоохранения в бедных странах. GNU Health принят Университетом ООН в качестве основной медицинской информационной системы.

Про Free Software Awards

Free Software Awards – ежегодная премия FSF за вклад в развитие СПО, основанная в 1998 году. С 2005 года введена вторая номинация, присуждаемая социально значимым свободным проектам. С 2001 по 2005 вручалась на FOSDEM (англ. Free and Open source Software Developers' European Meeting – Европейская конференция разработчиков свободного и открытого программного обеспечения). С 2006 года награды вручаются на ежегодном заседании членов FSF в Кембридже, штат Массачусетс.

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



Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
...простой советский юзер.

Блеск openSUSE

OpenSUSE попал на мой ноутбук неожиданно – из-за его «мичуринского» видео (см. прошлый номер) иные дистрибутивы не захотели запускаться. Однако прижился он не поэтому, а вследствие своих особенностей. Из которых отмечу самые для меня интересные.

Во-первых, модуль инсталлятора, ответственный за разметку диска и файловые системы – то, что потом менять сложнее всего. В openSUSE он на стадии установки позволяет сделать все, что ни придумаешь, вплоть до создания файловой системы ext4 «без журнала», определения любых субтомов для btrfs и монтирования tmpfs в любые уместные точки.

Во-вторых, менеджер пакетов *zypper*. Созданный по мотивам *apt* и *yum*, он выбрал в себя лучшие их черты: по простоте синтаксиса превзошел второй, по функционалу – первый. А «обертка» из YaST2 облакает его в форму элегантной графики.

В-третьих, сам YaST2 – универсальная среда для настройки всего и вся: оборудования, загрузчика, ядра, стартовых сервисов, сетевых соединений и служб, виртуальных машин...

И в-четвертых – «трехступенчатая стабилизация»: можно сидеть в текущем релизе, или идти вдоль верстовых столбов [Milestones] релиза грядущего, или, в рамках «текучки», применять лишь часть достижений светлого будущего, иначе доступных только в отдаленной перспективе.

Все это и определяет своеобразие openSUSE, делая его почти оптимальным дистрибутивом для настольной системы. Почему «почти»? А об этом – в следующем номере. alv@posix.ru

Сегодня мы рассматриваем:

- LibreOffice 3.5** 12
Теперь его разработчики не обязаны визировать свой код в Oracle или Sun, и LibreOffice повеселел, поздоровел и обзавелся проверкой правописания.
- Cinnamon 1.2** 16
Эта среда рабочего стола утешит многих разочарованных пользователей, предоставив слияние технологии Gnome 3 с удобством Gnome 2.
- Pandora** 18
Если вы искали игровую консоль, способную также запускать пакеты из ARM-ветки Debian, то обратились по адресу. Винтажные игры отлично себя на ней чувствуют.
- Explay Informer 701** 14
О правах root для пользователей устройств на Android говорилось много и с возмущением, а этот недорогой планшет без лишних слов их предоставляет. Вперед!
- VLC 2** 17
Им часто пренебрегают из-за отсутствия графического глянца, но во всем остальном VLC вполне преуспел. Хотя иконка его нам все равно не по душе.
- CrossOver** 19
Невыгодно приобретать лицензию на Windows всего из-за одного приложения? Вот вам выход – слой совместимости, позволяющий запускать Windows-приложения на Linux. Он превосходен.



Cinnamon 1.2 с. 16

› Cinnamon и Mint: прямое сочетание, перекрывающее все ваши Linux-потребности.



Pandora с. 18

› Носите игры 1980-х в кармане, на превосходной машинке Pandora.

Наш вердикт: Пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатипятибалльной шкале (0 – низшая оценка, 10 – высшая). Как правило, учитываются функциональность, производительность, удобство использования и цена, а для бесплатных программ – еще и документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту. Выдающиеся ре-



шения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов в GCC, но если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

AfterShot Pro

Разработчик: Corel/Bibblelabs
Сайт: www.corel.com/aftershotpro
Цена: \$99

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

» Проколы имеются, но AfterShot Pro – единственный вариант для серьезных занятий фотографией в Linux.

Рейтинг 9/10

LibreOffice 3.5

Этот релиз уже провозгласили «лучшим свободным офисным пакетом всех времен». Так ли это на самом деле? Взгляд **Бена Эверарда**.

Вкратце

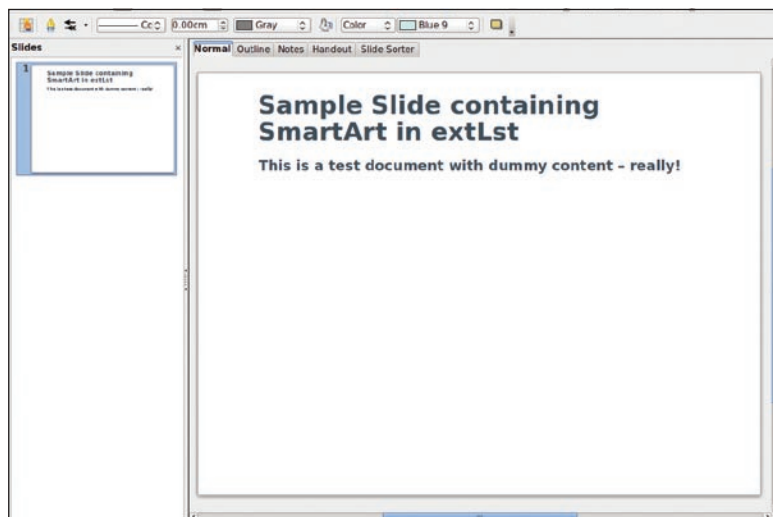
» Обновление самого популярного офисного пакета для Linux. См. также: *Gnome Office, KOffice, Google Docs* и *OpenOffice.org*.

Несмотря на солидную цифру в названии, это всего лишь третий релиз от The Document Foundation (TDF – организация, которая разрабатывает *Libre Office*): первым был 3.3, дабы совпадать с текущей версией *OpenOffice.org*, (OOo), от которой нынешний проект отделился. Тем не менее, за свою столь недолгую историю программа уже прошла большой путь. И самые значительные изменения коснулись не столько технической, сколько организационной части.

Хотя *OpenOffice.org* считался открытым проектом, формально он принадлежал Sun, а затем перешел к Oracle, так что направления его развития определял корпоративный тиран, вплоть до требования к разработчикам передавать права на код. TDF попыталась избавиться от этой дурной наследственности и сделать ПО дружелюбнее к своим создателям. Политика имела успех, и с момента отделения от OOo притянула более 300 новых разработчиков. Этот аспект проекта развивается параллельно с функциональностью, с каждой новой версией.

Присоединяйтесь

По случаю выхода версии 3.5 членам сообщества, внесшим наибольший вклад в ее создание, вручили особые значки за успехи в хакинге и за отлов ошибок. Если и вы задумаете пополнить многочисленные ряды разработчиков, за информацией обратитесь на <http://wiki.documentfoundation.org/Development>: там даже есть список задач, посильных для новичков.



» Рис. 1. В LibreOffice 3.4 объекты SmartArt открывались некорректно, оставляя вместо себя пробел. В версии 3.5 это было исправлено — см. рис. 2 на соседней странице.

documentfoundation.org/Development: там даже есть список задач, посильных для новичков.

Новый релиз *LibreOffice* принес заметные и практически повсеместные улучшения ПО. Обзоры офисных пакетов обычно начинают с редактора документов или электронных таблиц. Но мы сперва обратимся к графическим инструментам – именно там появилась столь долгожданная функция: фильтр импорта *Visio*, созданный в рамках Google's Summer of Code [Лето за кодом в Google, летняя практика для студентов].

Важность его в том, что раньше ни одно приложение Linux не работало с такими файлами нормально. И если обычные пользователи просто не воспринимали этот странный формат, то менеджеры проектов и те, чья деятельность связана с планированием, активно их использовали и были вынуждены искать ПО, соответствующее их запросам. Теперь же *LibreOffice* – а вместе с ним и Linux – стал жизнеспособной опцией в бизнес-сферах.

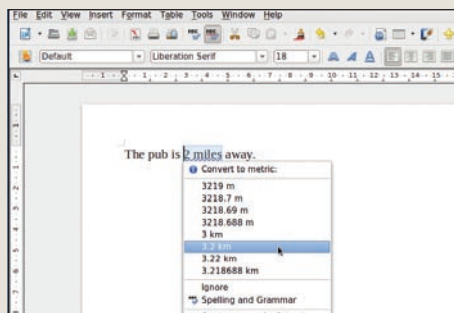
Еще одна важная прибавка к графическим составляющим пакета – импорт фигур SmartArt. Впервые представленные в *Microsoft Office 2007* как способ улучшить представление графики и текста, шире всего они применяются в *PowerPoint*. Так что, если вы были недовольны, открывая документы *Office* с графикой, *LibreOffice* вас спасет. Сколь бы мы ни желали, чтобы мир перешел на открытые форматы, этого пока нет, и улучшение поддержки проприетарных в этом пакете как нельзя кстати.

Если же презентации вы чаще создаете, чем открываете, новая функция встраивания видео или палитры в документ поможет вам распространять свои файлы с уверенностью, что они нормально отображаются на другом компьютере.

В помощь писателям

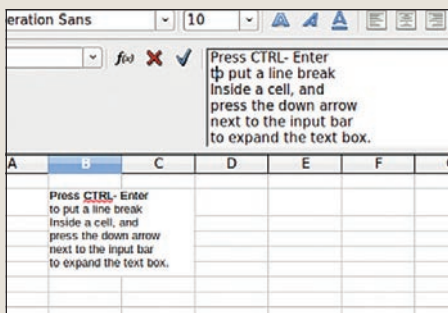
Здесь, в *Linux Format*, мы зарабатываем выпуском этого журнала, и охотнее всего занимаемся исследованием новых потре-

Свойства навскидку



» Конвертор единиц

С точностью до 10 мм, но ближе сделать не получится.



» Безграничные ячейки

Забудьте об однострочном вводе! Теперь ячейка с текстом может быть любого размера.

сающих технологий ПО открытого кода. Но, к сожалению, пока не изобретен компьютер, способный импортировать мысли прямо из головы, заодно преобразуя их в безупречный английский, нам придется еще и мучиться с опечатками и тому подобным. Проверка правописания (а она в *LibreOffice* неплоха), конечно, выручает, но это всего полдела. В версии 3.5 команда *LibreOffice* порадовала нас еще и проверкой грамматики, чтобы облегчить наши страдания, но хорошо ли она работает?

Ну, хорошая новость в том, что она не подчеркивает все сплошняком в вашем безупречном тексте, считая его якобы неправильным. Разработчики постарались максимально снизить количество ложных тревог, и текст выделяется только при явной ошибке. Следствием стали пропуски простейших грамматических нарушений. Когда мы в порядке эксперимента ввели следующий текст:

“Its a cold day. It a cold day. it’s a cold day. It’s an cold day”

он нашел всего две ошибки: неправильное употребление артикля “an” и отсутствие заглавной буквы [на самом деле, ошибка есть в каждом предложении, – прим. пер.]. И хотя лишняя перепроверка не во вред, наш выпускающий редактор Гэри (вычиты-

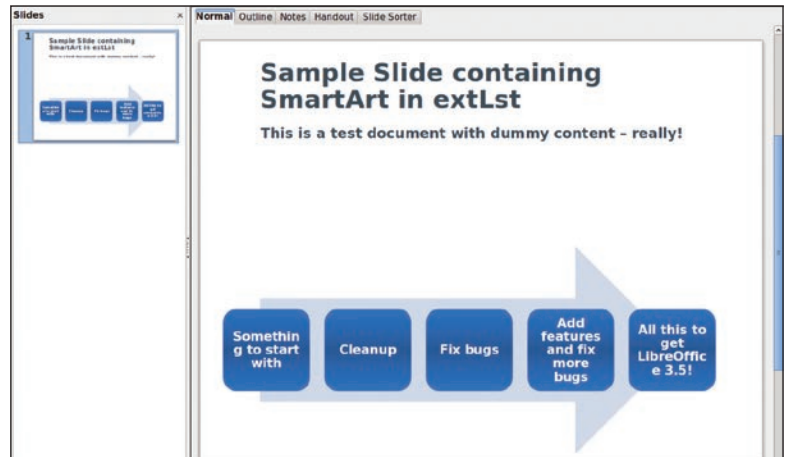
«Можете создать таблицу хоть на 10000 листов, если вы садист.»

вающий нашу сумбурную прозу и правящий грамматику) работы явно не лишится.

Этот инструмент не только проверяет грамматику, но и преобразует единицы измерения. Если опция включена, то при каждом вводе значения в распознаваемых системой единицах текст будет подчеркнут волнистой синей линией. Нажатием правой кнопки вы можете конвертировать их в другие поддерживаемые единицы. Чтобы включить функцию, выберите Инструменты > Менеджер дополнений и выделите словарь, затем щелкните по Свойствам и отметьте галочкой параметры, которые хотите конвертиро-

Apache Office

Пока *LibreOffice* штурмует вершину за вершиной, *OpenOffice.org* тоже не стоит на месте. Oracle передал права на продукт Apache Foundation, где и готовится к выпуску версия 3.4. Пожалуй, главной подоплекой этой передачи станет смена лицензии. Apache License, в отличие от текущей LGPL, не требует от третьих сторон публикации внесенных изменений. Такое решение может заинтересовать бизнес-владельцев, но раздражает ортодоксов свободного ПО.



► Рис. 2. Тот же файл, что и на рис. 1, но открытый в 3.5. На сей раз графика на месте, и презентация выглядит гораздо симпатичнее.

вать. Нам показалось, что некоторые найти непросто, но, возможно, дело в многофункциональности программы.

Внешний вид колонтитулов и система подсчета слов также переработаны в сторону удобства. Если вы пишете текст заданного объема, вам пригодится новое интерактивное окно подсчета: оно теперь обновляется по мере вашего печатания, и вы можете наблюдать, как уменьшается или увеличивается ваш задел при внесении изменений. Появились также новые функции улучшения раскладки и отображения шрифтов, чтобы документы выглядели более профессионально.

Сводя счеты

Виртуозы электронных таблиц оценят возможность делать сразу несколько выделений в фильтре. По сути, это способ создать простейшие фильтры ИЛИ. Чтобы воспользоваться данной функцией, выделите несколько ячеек и выберите Данные > Фильтр > Автофильтр. Таблица покажет вам выпадающее меню над каждой колонкой, где надо отметить элементы, которые вы хотите показать. Ранее эти действия можно было выполнить только с одним элементом, но теперь есть список, где их можно отметить сколько угодно.

Еще одно новшество в электронных таблицах – неограниченность правил условного форматирования. Условное форматирование – это применение разного стиля оформления (цвета, кегля и т.д.) в зависимости от содержимого ячейки. Можно, например, выделить все ячейки, значения которых превышают определенный порог или содержат определенное слово. В более ранних версиях к одной и той же ячейке допускалось применять не более трех условий; теперь их число не ограничено.

Мелкое, но полезное изменение – реализация многострочного ввода. Вам больше не придется тесниться в одной строке, вводя большие объемы информа-

ции в одну ячейку. Можете создать таблицу хоть на 10000 листов, если вы такой садист. Возможно, когда-нибудь, кому-нибудь, неведомо кому, она и пригодится – лишь бы информация из этих документов не понадобилась нам.

Пользователей *Base*, редактора баз данных, тоже не обидели, но им придется довольствоваться меньшим. Появился собственный фильтр для *PostgreSQL*, позволяющий прямо из документа подсоединяться к этому виду серверов. Также в этой версии возвращен инструмент онлайн-обновления. Он периодически проверяет наличие более новой версии *LibreOffice* и (если эта опция выбрана) скачивает ее.

Была проделана и большая внутренняя работа по чистке кода и локализации. Сейчас она не столь заметна, но непременно принесет плоды в будущем, поскольку упростит программистам исправление ошибок и добавление новых функций.

В общем, релиз достойный, с важными нововведениями и устраненными ошибками. Но если код станет еще лучше, а разработчиков еще больше, *LibreOffice* докажет нам, что это не предел – ждем не дождемся. LXF

LINUX Вердикт
FORMAT

LibreOffice

Разработчик: The Document Foundation
 Сайт: www.libreoffice.org
 Лицензия: LGPL

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

» Возможно, лучший свободный офисный пакет всех времен, но стремиться еще есть куда.

Рейтинг 8/10

Explay Informer 701

Андрей Боровский рассказывает о самом дешевом планшете под управлением Android 4.0.x на отечественном рынке.

Кратце

» Explay Informer 701 — бюджетный планшет, способный, однако, предоставить почти полноценную функциональность Linux. См. также: Onda V110 Deluxe.

Планшет Explay Informer и правда сравнительно недорог — в районе 4,3 тыс. руб., однако интересен не столько дешевизной, сколько тем, как легко его превратить к компьютер Linux, не теряя функциональности Android.

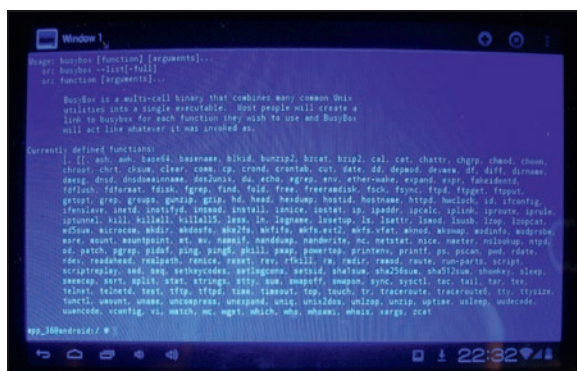
Аппаратная начинка и аксессуары

Сердце (и мозг) Explay Informer 701 — процессор VoxelChip A10, он же Allwinner A10 — содержит одно ядро ARM Cortex A8 и графическую систему Mali 400. Этот чип умеет воспроизводить видео в формате 1080p (FullHD) с достойным битрейтом (есть и приличествующий такому случаю разъем mini-HDMI), а также справляется со многими 3D-играми. Скажем, *Angry Birds* для него — не проблема.

В некоторых обзорах процессору VoxelChip A10 приписали ядро Cortex A10, на которое как бы намекает название чипа, но такого пока в природе нет. В сочетании с 512 МБ ОЗУ и 4 ГБ встроенного хранилища (расширяемого на 32 ГБ внешней картой microSD) устройство выглядит скромно — впрочем, как и его цена.

Разрешение фронтальной камеры — 1,3 Мпикс. 7-дюймовый емкостной экран, с разрешением 800×480 пикселей и пропорциями 16:9, обладает функцией автоматического поворота в зависимости от положения устройства. Подсветка равномерная и достаточно яркая. Поддержан мультитач на 5 касаний. Корпус выполнен из «несолидного» пластика, что уменьшило вес девайса. В целом сборка хорошая, без люфтов, при легком перекручивании не скрипит и не шатается. Корпус глянцевый, маркий. Кнопки утоплены, для снижения вероятности случайных нажатий.

» BusyBox в режиме root на Explay Informer 701.



» Explay Informer 701, общий взгляд.

Механические кнопки размещены на верхнем торце и легко доступны при просмотре видео в горизонтальном положении. Группы кнопок «Громкость» нет: на торец устройства вместо них вынесены управляющие кнопки «Меню», «В начало», «Esc», а для регуляции громкости в строку состояния ввели две сенсорные кнопки — своего рода «фирменная примочка» устройства. Почему их только две («громче» и «тише») и куда делась сопутствующая им обычно кнопка «отключить динамик», остается гадать. Виртуальная клавиатура Explay Informer 701 особых сюрпризов не преподносит и по умолчанию настроена только на язык системы (т.е. русский). Добавлять английский, без которого в Интернете туго, придется в разделе настроек клавиатуры.

К зарядному устройству и к компьютеру планшет подключается через один и тот же разъем, а зарядка идет в обоих случаях — удобный прием, который могли бы взять на вооружение и другие производители. Увы, аккумулятор при интенсивной работе садится довольно скоро, часа за 2–3.

В комплект поставки также входят зарядник, наушники (с вариантами насадок и не самым скверным воспроизведением), кабель для подключения устройства к ПК и защитный чехол с прорезью в виде кармана — фасон не особо модный и удобный (тот же Texet tm-7025 поставляется с гораздо более стильной «одежкой»), но, как говорится, за такие деньги и на том спасибо.

Удивляет отсутствие «бумажного» руководства пользователя (в самом устройстве его можно найти в виде PDF-файла).

ОС и функциональность

Предназначенную специально для планшетов версию Android 3.x разработчики почему-то обошли стороной и ставят на устройства либо Android 2.3.x, либо 4.0.x.

Только факты

- » Операционная система: Android 4.0.3
 - » Процессор/чипсет: VoxelChip A10, 1 ГГц
 - » Оперативная память: 512 МБ DDR2
 - » Встроенная память: 4 ГБ
 - » Экран: 7", 800×480
 - » Тип экрана: TFT, емкостной, мультитач
 - » Автоматическая ориентация экрана
 - » Поддержка Wi-Fi 802.11n
 - » Фронтальная камера, 1,3 млн пикс.
 - » Встроенный динамик
 - » Встроенный микрофон
 - » Акселерометр
 - » Поддержка форматов: AAC, WMA, WAV, OGG, FLAC, APE, MP3, MKV, H.264, MOV, MP4
 - » Размеры (Д×Ш×Г) 186×145×12 мм
 - » Вес 340 г
 - » Bluetooth отсутствует
 - » GPS отсутствует
- Характер стойкий, корпус пластмассовый, к пользователям суров, но не беспощаден.

Вообще, говоря о Informer 701, чувствуешь себя парашютистом с нераскрывшимся парашютом. Как в детском анекдоте: «вижу под собой стог сена, это хорошо. В нем вилы – это плохо...» и т.д. Устройство поставляется с рутованной прошивкой (т.е., подключаясь к устройству через терминал, вы сразу получаете права root). Хорошо! Но некоторые пользователи жалуются, что после *n*-го выключения питания (число *n*, похоже, случайное и колеблется в интервале 1–1024) автоматически возвращаются заводские настройки. Права при этом не исчезнут, зато «забудутся» установленные приложения, пароли Wi-Fi, учетные записи почты. Излишне говорить, что это плохо. Зато разработчик прилагает сразу несколько официальных прошивок с подробными инструкциями по прошиванию, и это хорошо. Во всех прошивках есть Google Market, или, по-новому, Google Play. Это хорошо. Но нет распознавания голосового ввода, а это плохо. Уверенный прием сигнала Wi-Fi – для планшетов такого уровня очень хорошо. Нет Bluetooth... но я бы не сказал, что это плохо. Я, кстати, вообще не понимаю, зачем Android-планшетам Bluetooth. Вибросигнала и камера на тыльной стороне тоже нет. Есть кабель для подключения устройства к ПК,



» Onda VI10 Deluxe.

(хотя внешний модем и менее удобен, чем встроенный модуль GSM/3G).

Воспроизведение видео в том из форматов, декодирование которых реализовано на аппаратном уровне, выполняется плавно и энергии потребляет немного, а с помощью программного кода – приводит к пропуску кадров и заметному повышению расхода энергии (при этом девайс прилично нагревается). Под фирменной пленкой Explay расположен еще один слой защитной пленки, который (ура!) не препятствует работе с экраном и не портит качество изображения. Сенсор послушен и фиксирует нажатие именно там, где нажато – большой плюс для устройства с маленьким экраном, где набор текста при недостаточно точном сенсоре может превратиться в пытку.

Здесь можно бы заявить, что Explay Informer 701 предоставляет замечательные возможности за свои деньги – и поставить точку, но на самом деле это далеко не все.

Экспериментаторам

Мой интерес к этому устройству был вызван, прежде всего, такими факторами: наличием рутованной прошивки, открывающей массу возможностей для тех из нас, кто видит в ОС Android еще и Linux, и новейшая версия Android, которая, помимо прочего, открывает широчайшие возможности для написания программ на C и C++. Все это, при невысокой цене (можно особо не плакать, если в итоге очередного эксперимента планшет превратится в бесшумный кирпич), делает Explay Informer 701 весьма подходящим устройством для желающих разобраться во внутренностях Android, железа ARM, Linux для ARM и т.п.

С чего начинается исследователь нового явления природы (будь то ранее неизвестное животное, звезда или химическое соединение)? С установления возможного родства наблюдаемого явления с уже известными. На рынке у Explay Informer 701 есть не то что родственник, а, можно сказать, брат-близнец Onda VI10 Deluxe (фото с сайта www.kaskus.us). Фактически это одно и то же устройство под разными брендами. Так что многие ресурсы, пригодные для Onda, подойдут и для Explay.

Перейдем к программной начинке устройства. Встроенная оболочка командной строки (к ней можно получить доступ из эмулятора терминала или подключив устройство к ПК через утилиту *adb*) обзавелась функцией автозавершения вводимой строки по нажатию на клавишу табуляции. Тот, кто экспериментировал с ранними версиями Android, поймет, сколь удобнее стало работать с новой оболочкой. А если удобств встроенной оболочки вам мало, установите полноценную *bash* – сначала потребуются установить *BusyBox* для Android, но этот набор полезных утилит командной строки всяко стоит установить.

Для комфортной работы с Android в стиле Linux нужна специальная файловая оболочка. Я рекомендую две: ES File Explorer и Ghost Commander. Обе не только весьма удобны для работы с файлами и содержат кучу дополнительных утилит, но и обладают функциями для работы в режиме root, в том числе перемонтирования системных файловых разделов с правами записи.

Можно установить и сервер SSH, которых для Android пишется уже несколько (например: <http://code.google.com/p/droidssh/>). Есть для Android и консольная оболочка *Midnight Commander*. В силу технических причин запустить эту оболочку в эмуляторе терминала Android нельзя, зато можно запустить ее в терминале настольной Linux (или Windows), подключившись к устройству Android по SSH.

Итак, что же мы получили? Без особых ухищрений – не сложнее установки дистрибутива Linux в специальный раздел – карманное устройство Android можно превратить в пусть и маломощную, но практически полноценную Linux-систему, пригодную для самых разных целей, начиная с демонстраций и изучения разработки программ для процессора ARM и заканчивая созданием экономичного и бесшумного сетевого сервера для домашних нужд. Именно этим, на мой взгляд, более всего интересен Explay Informer 701. **LXF**

«Замечательные возможности за свои деньги, но это не все.»

а вот кабель для подключения внешних устройств USB в поставку не вошел – решив подключить к девайсу флэшку, сразу идите покупать переходник. По отзывам ряда пользователей, к Explay Informer 701, помимо флэшек и клавиатур, можно подключить USB-модем, сделав его полноценным мобильным интернет-устройством



» Ghost Commander.

LINUX **Вердикт**
FORMAT

Explay Informer 701

Разработчик: Explay
Сайт: www.explay.ru/catalogue/tablet-pcs
Цена: ок. 4300 руб.

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

» Недорогое устройство получилось вполне добротным.

Рейтинг 7/10



Cinnamon 1.2

Очередное рабочее окружение... правда, Джонатан Робертс полагает, что фанатам Gnome 2 на сей раз есть чему порадоваться.

Вкратце

» Гибкое рабочее окружение, ответственное от Gnome 3. См. также: Unity, Gnome 3, Xfce.

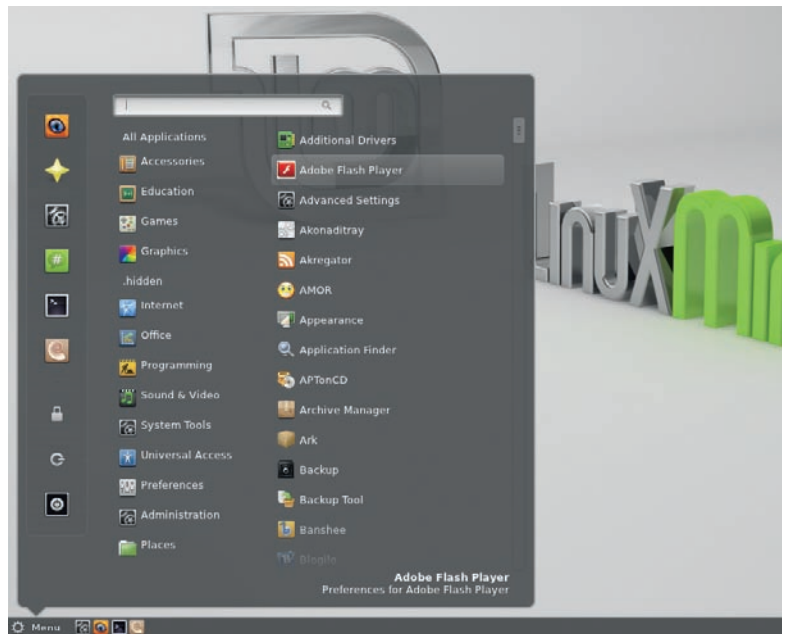
История эта уже приелась. Жил-был Gnome 2, и пользователи его знали, и любили в нем работать, и все их устраивало. А потом явились Gnome 3 и Unity, каждый со своим уставом по дизайну интерфейса, и поломали привычный уклад – а люди почему-то не пришли в восторг от насильственно навязанных перемен.

С тех самых пор таили они надежду на избавление от этой беды. И последний из явившихся спасителей – проект Cinnamon. Эта вторая попытка команды Linux Mint создать рабочее окружение – бальзам на раны пользователей, переметнувшихся к нему из-за разочарования в других крупных дистрибутивах.

Сразу же после загрузки становится ясно, что Cinnamon создан по образу и подобию рабочего стола Mint из последних релизов. Единственная панель, вместе с иконками быстрого запуска, списком окон, системным лотком и общим меню, расположена внизу экрана.

К избранным приложениям можно добраться за один щелчок, и там же, прямо под ними, блокировка экрана, завершение сеанса и выключение. И все они, в отличие от кое-какого другого рабочего стола, доступны всегда.

Остальные установленные программы находятся справа, в классическом меню, упорядоченном по категориям. Досадно, что не обходится без прокрутки, если надо видеть полный список приложений, даже в сравнительно малонаселенных категориях, но мы не считаем это особенно боль-



» Приятно-знакомая внешность Cinnamon, наверное, придется по нраву линуксоидам, разочарованным в Gnome 3 и Unity.

шой проблемой, ведь программу несложно добавить в избранные всего лишь правым щелчком мыши.

Знакомое чувство

По сравнению с Gnome Shell и Unity, меню воспринимается как более привычное. Да и быстрое: ведь оно запускается при помощи клавиши Meta (Windows) и дает возможность набора имени программы для запуска, как и в двух вышеупомянутых окружениях.

Графический конфигуратор Cinnamon позволяет менять тему оформления и добавлять новые расширения – чего определенно не было в первоначальном релизе Gnome Shell. Учитывая то, что проект новый, количество доступных тем, дополнений и апплетов, доступных на официальном сайте Cinnamon весьма и весьма впечатляет.

Установку не помешало бы и улучшить, поскольку пока что от пользователя требуется распаковывать скачанные файлы в определенные папки. Так что в этом плане, Gnome Shell и Gnome 3 пока дают ему фору.

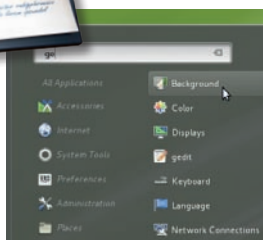
В общем и целом, мы полагаем, что Cinnamon обладает неплохим потенциалом. Нам понравилось, что в нем все знакомо и от пользователя ничего нигде

не прячут. До настроек добраться легко, и интуитивно понятно, какие окна нужно открыть, не переходя в отдельный режим.

Правда, многое еще нужно довести до ума, чтобы Cinnamon стал по-настоящему первоклассным рабочим столом. Было бы замечательно, к примеру, иметь способ автоматической установки расширений, и чтобы поиск в меню работал как надо; ну и чтобы всякие мелкие недостатки, такие как неудобное расположение меню и «торчащий» горячий угол, были устранены. **LXF**

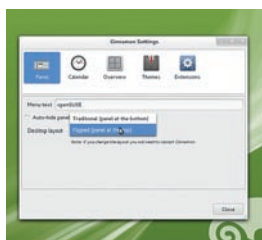


Свойства навскидку



Причуды поиска

Наберите в нем 'CM', и первым в результатах окажется фон, потому что именно там вы можете сменить обои.



Выбор — это хорошо

Вы легко можете переключиться на другое оформление, более напоминающее Gnome 2, в меню Настройки.

LINUX FORMAT Вердикт

Cinnamon 1.2

Разработчик: Linux Mint
Сайт: <http://cinnamon.linuxmint.com>
Лицензия: GPLv2

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

» Очень многообещающий проект. Будем надеяться, он получит необходимое внимание и доработку.

Рейтинг 7/10



VLC 2.0

Грэм Моррисон изучает крупное обновление медиа-плеера, недавно признанного лучшим по результатам голосования на TuxRadar.

Вкратце

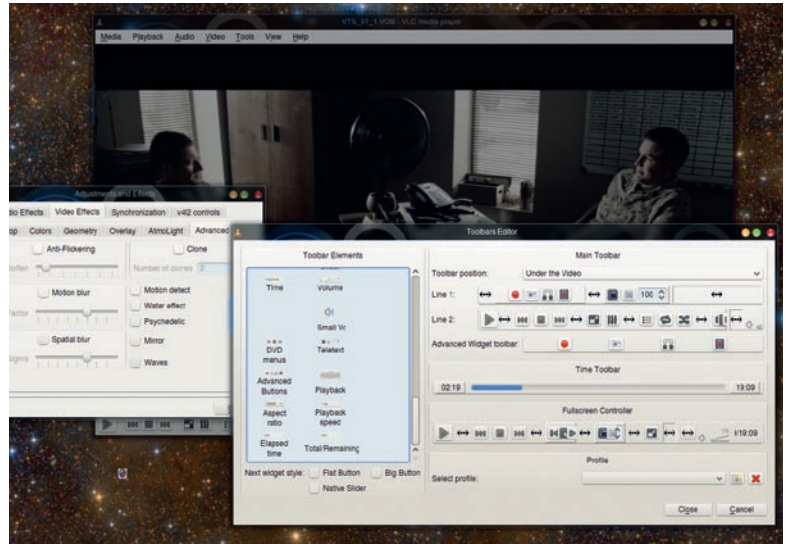
» Медиа-плеер, до которого всем остальным еще далеко. VLC проигрывает все, что угодно, и в лучшем виде. Или попробуйте *Xine* и *Mplayer*.

Несмотря на то, что корнями своими VLC уходит в сеть, о чем можно догадаться по его полному названию (Video LAN Client), этот потрясающий медиа-плеер снискал себе популярность у любой публики, какую бы ОС они ни предпочитали. И все потому, что он легко устанавливается и проигрывает почти все, что угодно. С VLC у вас более всего шансов открыть любой аудио- или видеофайл. Такой успех вывел программу на новый уровень разработки: с каждым новым релизом улучшались характеристики производительности, добавлялись новые функции, кодеки и медиа-форматы. А поскольку VLC – ПО открытого кода, пользователям Linux доступны те же обновления и тот же функционал, что и нашим собратьям Windows и OS X. Разве что... не совсем.

Выход второй версии – важный шаг, но одно из самых ярких новшеств – единый интерфейс окна для OS X. Обидно: ведь это как раз не помешало бы и нашей родной версии.

Муки интерфейса

Выполнить многие стандартные задачи нелегко. Если вы не знаете, как реализована та или иная функция и есть ли она вообще, то через графический интерфейс понять это невозможно. Лучший пример – просмотр локального медиаконтента UPNP. Для многих из нас UPNP – самый доступный способ загрузки контента с сервера. Однако вы не найдете его ни во вкладке Открыть, ни в Главном



» Увеличение с 224 до 176. Увеличение, стоп. Наезд, стоп. Растянуть, выровнять по правому краю, стоп.

меню, ни в многоуровневых панелях общих настроек. Вместо этого, столь необходимая функция оказывается в меню Плейлист – которое вы открыли бы, только решив создать свою собственную галерею файлов. И так во всем. Хотите ли вы проверить, что DTS идет через цифровой выход, или недоумеваете, почему по умолчанию выставлена половинная громкость – везде вам придется бороться с непостижимой логикой графических решений. Пользовательский интерфейс был расширен, чтобы вместить многочисленные функции VLC, и в результате выглядит путаным и неуклюжим. Обновить дизайн – это еще полдела, VLC требуется новый подход во всем остальном.

Но внешность – все же не главное. В других отношениях VLC нет равных. Обратной стороной этой неудачи с графическим интерфейсом является то, что на любом ваше пожелание найдется нужная функция, пусть найти ее будет и непросто. Чего мы только ни делали – и автозапуск из командной строки, и футбол в live-режиме, изображаемый как ASCII-текст, и просмотр местной телепрограммы. Есть новый web-интерфейс, доступный через меню Вид > Добавить интерфейс – он сразу дает вам контроль над воспроизведением. Переработано аппаратное ускорение как для аудио, так и для видеофайлов (теперь реализовано субпиксельное смещение на видеокарте), а также отображение суб-

титров. Капитально переделаны поддерживаемые кодеки и форматы, добавлено несколько профессиональных кодеков, а устаревшие форматы обновлены. Например, декодирование H.264, MPEG-4/Xvid и WebM стало многопоточным, и мы обнаружили, что воспроизведение файлов H.264MKV с высоким разрешением на нашей карте Nvidia теперь загружает процессор на 20 % меньше, чем прежние версии.

Качество воспроизведения несравненно, и версия 2.0 решительно увеличила отрыв от ближайших конкурентов. VLC сумеет справиться практически с любым форматом, и сделает это превосходно. Вот бы кто-нибудь добил интерфейс... **LXF**

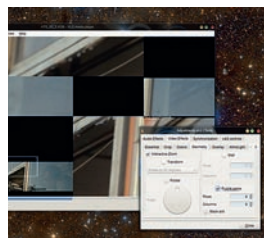


Свойства навскидку



Web-контроль

Спрячьте свой Linux ПК позади телевизора и управляйте воспроизведением через новый web-интерфейс.



Игры

К вашим услугам огромный выбор чудесных и удивительных игр, но «коллаж», пожалуй, самая странная.

LINUX FORMAT Вердикт

VLC 2.0.0

Разработчик: VideoLAN Project
Сайт: www.videolan.org
Лицензия: GPL/LGPL

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

» VLC – блестящий пример способностей свободного ПО.

Рейтинг 9/10

Pandora

Хранилище несчастий или компактный генератор Linux-позитива? Дэвид Хейворд открывает ящик Пандоры, чтобы разобраться.

Вкратце

» Мечта ретро-геймеров Linux воплотилась в маленькую аккуратную коробочку.

Только факты

- » **Процессор:** A8 600 МГц ARM
- » **Экран:** 800×400 LCD, сенсорный, резистивный
- » **ОЗУ:** 512 МБ
- » **Память:** от 512 до 64 ГБ
- » **Функционал:** аппаратный ускоритель PowerVR 3D, TV- выход, USB, противоударный корпус, ядро 2.6, вес 335 г с аккумулятором, размеры 140×83,4×27,5 мм

Пару лет назад команда талантливых ребят решила создать карманное игровое устройство с открытым кодом; и, в конце концов, мечта Крейга Ротвелла [Craig Rothwell], Мехмета Фатиха Килича [Mehmet Fatih Kilic] и Михаэля Мрозека [Michael Mrozek] воплотилась в Pandora. Это инженерное решение впечатляет. В коробочке размером не больше Nintendo DS уместились процессор ARM Cortex-A8 600 МГц, оперативная память DDR-333 SDRAM на 256 МБ, встроенный Wi-Fi, Bluetooth, сенсорный экран, два аналоговых джойстика, набор игровых кнопок и, на удивление, даже QWERTY-клавиатура с 43 клавишами. А в сердце Pandora – специальная версия Ångström Linux, которая навряд ли придется по вкусу новичкам Linux-сообщества; но, освоившись, вы оцените этот дистрибутив по достоинству.

Терпение – это добродетель

При первом запуске Pandora программа-установщик заставит вас выбрать имя пользователя, пароль, имя системы и часовой пояс и, наконец, рабочий стол. Как наладонник и игровая приставка в одном флаконе, Пандора поставляется и с обычным Linux для Xfce, и с упрощенным Ptemu, «заточенным под игры»: только кнопки и иконки. Вы можете установить один из них по умолчанию, а можете и переключаться между обоими, просто нажимая прилагаемым стилусом на иконку нужного интерфейса. Вообще-то и большинство основных рабочих операций выполняется именно так.

Тем не менее, чтобы вполне освоить и приручить Pandora, придется поглуб-



» Карманное игровое устройство на основе Linux, плюс все остальное.

же копнуть Ångström и, как часто бывает с малыми Linux-дистрибутивами, довести до ума пару приложений, чтобы они работали как надо.

Тут надо отметить, что Linux-часть в Pandora хорошо документирована, а справка доступна через форумы Open Pandora; так что, повозившись часок-другой с Ångström и Pandora, вы сможете скачивать и обновлять приложения, как маститый специалист.

Что наша жизнь? Игра!

Поклонников игр в ретро-стиле при виде Pandora наверняка прошибет ностальгическая слеза. В недрах репозитория Pandora таятся тонны эмуляций классических 8-битников в более современные 64-битные консоли. Есть авторские игры от бесчисленных разработчиков, а также часть привычных х-игр, портированных на ARM. Добавьте к этому джойстики Pandora – и вы в прошлом. Действительно, Pandora справляется с этим просто великолепно, несмотря на свои скромные размеры, без единой наклейки или видимого сбоя. Достойная демонстрация способно-

стей 600-МГц A8s! Pandora – это продукт Linux, любовно изготовленный преданным сообществом. И пусть он не заменит вам ваш планшет Android или какой-нибудь именитый игровой наладонник, но заполняет нишу между ними, совмещая рабочие возможности Linux с игровыми. Надежный и оригинальный, Pandora может украсить вашу мобильную жизнь. LXF



» Многочисленные порты – одно из главных достоинств Pandora.

LINUX **Вердикт**
FORMAT

Pandora

Разработчик: OpenPandora
Сайт: <http://openpandora.org>
Цена: €440 (примерно £373)

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

» Впечатляющий наладонник, хотя и на любителя, и довольно дорогой.

Рейтинг **7/10**

CrossOver 11

CrossOver Office и *CrossOver Games* умерли – да здравствует *CrossOver 11!*
 Расследование ведет Грэм Морриссон.

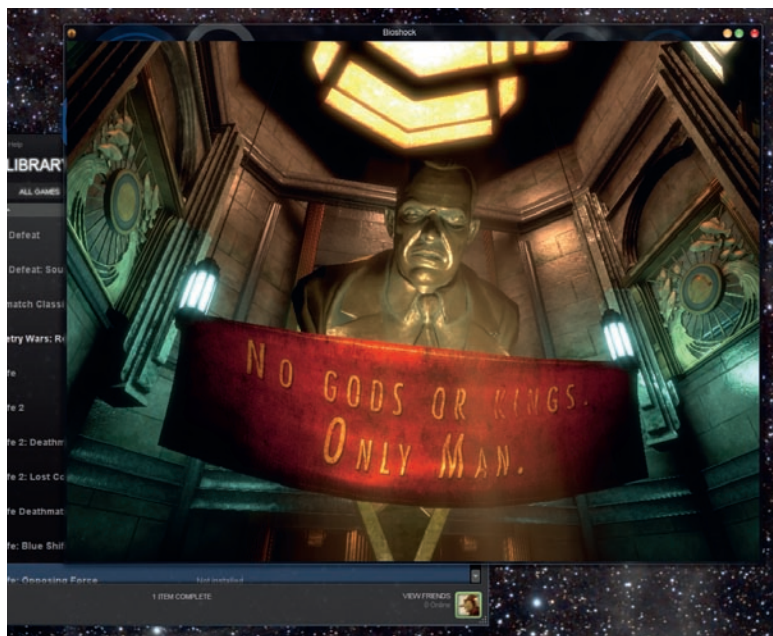
Вкратце

» Запускайте приложения Windows без всяких виртуализаций или копий. См. также *Wine*.

CrossOver – очередной «не эмулятор». Это «уровень совместимости», воссоздающий библиотеки Windows настолько, чтобы приложения Windows ощущали себя в родной среде. На заре развития *CrossOver* это означало возможность запустить *Internet Explorer*, чтобы попасть в свой интернет-банк, или работать с документами *Word* от коллег. Но широкое распространение свободного ПО сделало эти задачи неактуальными, и *CrossOver* перестроился – сначала разделился на игровую и офисную версию, а сейчас вновь собрал их в один продукт.

Как и прежде, мы видим симбиоз *CrossOver* и *Wine*: все значительные обновления из данного коммерческого релиза перейдут в *Wine 1.4*. Как следствие, в обоих налицо улучшения производительности и совместимости. Существенные различия наблюдаются лишь в установке и настройке приложений – *CrossOver* в этом плане значительно проще.

Сначала мы попробовали установить 32-битный Deb-пакет на Ubuntu 11.10, просто дважды щелкнув по нему в менеджере загрузок *Firefox*. После переподключения единственной помехой оказалось отсутствие подписей у значков приложений: мы не стали гадать, которое же является утилитой установки, и прибегнули к 64-битному пакету Arch. Здесь потребовалось просто запустить главный сценарий в корне, а все остальное выполнялось автоматически. Как и в предыдущих версиях, *CrossOver* использует набор профилей для установки необходимых вам при-



» Если у вас стоит более ранняя версия *CrossOver Office* или *CrossOver Games*, ваши приложения и игры в ходе установки будут объединены.

ложений Windows. С его помощью можно запустить любой исполняемый файл Windows, но гарантиями корректной работы являются все-таки эти самые профили. Каждый из них имеет бронзовую, серебряную или золотую медаль за совместимость, и вы можете посмотреть уже имеющиеся через установщик или закачать новые с сайта *CrossOver*.

Золотая лихорадка

Нашей исходной попыткой стала установка *Microsoft Office 2010*, но она не удалась из-за множества ошибок. Мы не первые, кто столкнулся с этой проблемой, и поскольку совместимость была помечена как «бронзовая», мы решили попытаться счастья с каким-нибудь «золотым» приложением.

Через несколько минут, закачав шрифты, *DirectX*, парсер *Microsoft XML* и *Adobe Flash*, мы получили готовый к работе *Internet Explorer 6* одним щелчком. Следующим тестом стала установка клиента *Steam*. И опять же, буквально за пару щелчков, мы его запустили и вовсе скачивали игры из нашей библиотеки. Сначала попробовали *Portal*, но возникла проблема с драйверами *Nvidia*. Поскольку наша Arch-система 64-битная, многие игры на *CrossOver* требуют установки 32-битного пакета утилит. Для пользователей Arch по-

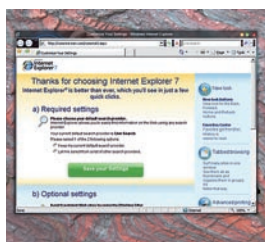
дойдет *lib32-nvidia-utils-beta* из репозитория AUR. В итоге, *Portal* пошел идеально, равно как и *Half Life*, *Day of Defeat* и даже *Bioshock*, у которого теперь тоже «золотой» статус. Нормальная работ *Skyrim* тоже обещана, но испытать его нам не довелось. Характеристики визуализации и производительности по сравнению с прошлыми версиями улучшились; и даже если с ультрасовременными приложениями Windows *CrossOver* не очень справляется, то поиграть в игры Windows нескольких лет давности вам удастся на все сто. **LXF**



Свойства навскидку



Мгновенная установка
 Если все файлы есть в открытом доступе, *CrossOver* скачает зависимости и все установит автоматически.



Совместимо с Windows
 Многочисленные приложения, которые были выпущены несколько лет назад, заработают без проблем.

LINUX FORMAT Вердикт

CrossOver 11

Разработчик: CodeWeavers Ltd
 Сайт: www.codeweavers.com
 Цена: £32

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

» Позволит установить сотни игр и приложений, а также поддерживать ценную ОС.

Рейтинг 9/10



ANDROID MARKET УЖЕ НЕ ANDROID MARKET

Магазин меняет имя

Android Market преобразован в ориентированный на контент сервис Google Play.

Компания Google объявила о введении в строй универсального сервиса Google Play, который заменил собой магазин-каталог приложений Android Market, расширив его средствами доставки контента. Кроме распространения приложений и игр для платформы Android, сервис Google Play прежде всего нацелен на предоставление средств для загрузки на устройства пользователей контента, такого как музыка (20 тысяч бесплатных композиций и несколько миллионов платных), электронные книги, фильмы, в том числе недавно вышедшие на экраны, и материалы в формате HD (стоимость фильмов от 99 центов).

» Изменив название и приоритет своего каталога приложений с ПО на контент Google сделала правильный выбор.



Интересной особенностью Google Play является привязка к учетной записи пользователя, что позволяет использовать выбранный контент на любых устройствах, к которым пользователь имеет доступ. Google Play выступает в роли облачного хранилища, предоставляющего доступ

к контенту в любое время из любой точки Сети. К сожалению, магазин контента пока открыт только для пользователей США, Канады и некоторых других стран, в число которых Россия не входит. Раздел с приложениями для платформы Android не имеет таких ограничений. Функциональность и протоколы взаимодействия Android Market не изменились – сервис был лишь переименован и расширен. Для пользователей приложения Android Market на телефонах и планшетах все осталось как прежде – по крайней мере, до установки обновления, которое уже успешно распространяется для некоторых моделей телефонов.

НА ПУТИ К СВОБОДНОЙ ПРОШИВКЕ

Replicant – не репликант!

Недоделанная свободная Android-прошивка Replicant доступна для Galaxy S.

Проекту Replicant, в рамках которого предпринята попытка создания полностью открытого варианта платформы Android, не содержащего проприетарных компонентов и драйверов, объявил о публикации нового набора установочных образов на базе Android 2.3 и появлении сборки для смартфонов Samsung Galaxy S (ранее поддерживались только телефоны Nexus one и Nexus S).

По сравнению с оригинальной прошивкой, в Replicant для Galaxy S была заменена достаточно большая порция проприетарных компонентов, включая видеодрайверы для GPU PowerVR, двоичную прошивку для Wi-Fi, библиотеки для работы с GPS, компасом, web-камерой, радиоинтерфейсом и модемом.

Статус поддержки Galaxy S выглядит следующим образом:

- » работают: 2D-графика, Wi-Fi, Bluetooth, звук, 3G, телефония, FM-радио, акселерометр;
- » не работают или работают нестабильно: 3D-графика (только минимальная поддержка), GPS, компас, web-камера.



» Полностью свободная прошивка для Android-устройств уже реальность. Но зачем?

Replicant

Дополнительно можно отметить выпуск компанией Google новой версии эмулятора Android, в котором существенно увеличена производительность за счет GPU-акселерации и двукратного ускорения эмуляции инструкций CPU. В эмуля-

торе теперь полноценно поддерживается OpenGL ES 2.0, что позволяет без проблем тестировать OpenGL-игры. В новой версии также расширен спектр эмулируемого оборудования: например, обеспечена поддержка эмуляции дополнительных аппаратных сенсоров и систем ввода с поддержкой мультитач. В будущих версиях ожидается поддержка Bluetooth и NFC.

А надо ли оно обычным пользователям? Вопрос открыт.

Android и патенты

За свою недолгую жизнь платформа Android из гадкого утенка превратилась в прекрасного лебедя. Именно так можно охарактеризовать взрывной рост популярности и закрепление доминирующих позиций данного творения Google. Компании, до этого считавшие себя титанами, потеснились, уступая место зеленому дроиду, но не сдались на милость «корпорации Добра», а решили всеми своими силами «завалить» Android.

И, похоже, сейчас у них более всего шансов это сделать. Microsoft и Apple действуют не сообща, но их действия носят характер скоординированной атаки. Сейчас очень многое зависит от объективности и непредвзятости (а также честности и неподкупности) судов США. В случае, если Google проиграет судебную битву двум гигантским противникам, это будет означать конец платформы Android в ее нынешнем виде...

ЯБЛОКО НОРОВИТ СЪЕСТЬ АНДРОИД

Факт – Apple угрожает Android сильнее, чем Microsoft

Apple пытается заблокировать продажи Android-устройств в США.

Компания Apple стоит на пороге достижения своей главной текущей цели – ликвидации Android как основного конкурента и полного итогового доминирования на мобильном рынке США. Произойти это может в том случае, если будет выигран патентный иск Apple против Motorola. Патент, фигурирующий в деле, печально известен как «патент на мультитач» (патент США номер 7,479,949). Патент защищает не частный случай, описываемый конкретным алгоритмом, но абстрактное описание всех алгоритмов, имеющих отношение к использованию мультитач-экранов, в особенности принцип отсеивания «неточностей и дрожания» при распознавании полулинейных жестов.

Дело ведет известный судья Ричард Познер [Richard Posner], который уже присудил Apple победу в январе этого года по итогу правового анализа формулировок патентного иска. Также в решении от 29 марта этого года судья постановил, что интерпретации Apple технической языковой конструкции иска в основном верны, и наказал юристов Motorola Mobility за несостоятельные и повторяющиеся доводы. Юристы Motorola настаивали на том, что патент должен быть действителен только для жестов, сделанных под углом в 27° (так называемый «фактор допустимости ошибок»), обсуждаемом в примере реализации технологии, однако судья три раза отвергал эти аргументы на основе того, что угол в 27° приводится в спецификации просто как частный пример.

Мартовское решение по иску открывает дорогу судебному запрету на продажу всех мультитач-устройств производства Motorola, и скорее всего за этим последуют запреты для компаний Samsung Electronics и HTC. Вероятность запрета подтверждает также и тот факт, что Apple не желает лицензировать данный патент, что дало бы возможность просто выплачивать лицензионные отчисления со стороны этих компаний за право использования интеллектуальной собственности Apple.

Возможность возникновения подобных ситуаций является прямым следствием того, что Бюро США по патентам и торговым знакам позволяет патентовать небрежно сформулированные, двусмысленные и ту-



➤ Android вынужден отбиваться от копирастов. Доколе?

маные описания программ, а далее судебная система США придает им легальную силу, с деструктивными и мешающими свободной конкуренции последствиями.

«Бюро США позволяет патентовать двусмысленные описания.»

Какие возможности существуют в данной ситуации у производителей Android-устройств? Если Apple получает решение о полном запрете на продажи, то у нее будет полный контроль над тем, позволить ли лицензирование по завышенным ставкам или просто дать умереть продажам Android по всей территории США. В любом случае, возможностями Android-производителей будут:

- сужение области технологий, подпадающих под действие патента в аналогичных исках Samsung/HTC;
- передача дела в более высокие федеральные судебные инстанции, что лишит юридическое действия технический анализ судьи Познера;
- нахождение достаточного количества доказательств фактов более раннего использования данных технологий, чтобы убедить патентное бюро в недействи-

тельности этого патента с широкой областью действия;

- победа иска против Apple с аналогичной силы решением не в пользу Apple, чтобы вынудить ее пойти на кросс-лицензирование.

Упомянувшийся в Новостях Флориан Мюллер, автор известного правового блога fosspatents.com, также уделяет достаточно внимания данному патентному иску. В своем блоге Мюллер подробно разбирает терминологию патента относительно того, как некоторые пользовательские жесты реализуются практически, согласно формулировке патента. Apple в своем заявлении объясняет: «Обозначив проблемы, связанные с неточными жестами, решив их и успешно встроив результаты в коммерческий продукт, Apple имеет право на получение плодов своих инноваций с помощью патентной защиты, а широкие массы пользователей выиграют от раскрытия изобретения Apple». Однако, замечает Мюллер, широкие массы пользователей выиграют от раскрытия плодов инноваций Apple только после истечения срока действия патента, то есть в 2028 году, и судя по всему, под этой формулировкой Apple подразумевает, что публика «выиграет» от полного запрета продаж смартфонов на базе Android в течении пятнадцати лет. **LXF**



Борьба за права root

Евгений Балдин разбирается, что можно и чего нельзя вытворять со своим телефоном на Android. Последуем за ним...



Наш эксперт

Евгений Балдин
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Android — это Linux. Кто не верит, тот может скачать с <http://source.android.com/> исходники и убедиться в этом лично. Их можно даже скомпилировать (доказано Суапоген(mod)). Вы хотите модифицировать прошивку своего аппарата, дабы полностью управлять им? У вас точно нет более важных и интересных дел? Давайте сначала определимся, надо ли вам это.

Что такое root

Да, собственно говоря, ничего особо партизанского в этом понятии нет. Права root для Android ничем не отличаются от прав суперпользователя в GNU/Linux. Если ты root — то ты можешь сделать со своей системой все, в том числе и уничтожить ее.

По умолчанию — скорее всего, именно благодаря последней возможности — «неразумным» пользователям Android данная возможность производителями не предоставляется. Более того, попытки получения этих прав в большинстве случаев всячески блокируются или, как минимум, не одобряются.

Следует, правда, отметить, что подобная точка зрения, к счастью, постепенно меняется. Например, HTC официально позволяет разблокировать загрузчик для последних версий своих телефонов (<http://htcdev.com/bootloader/>), в недрах той же Samsung получил пристанище лидер проекта СуапогенMod Стив Кондик [Steve Kondik], и даже Sony предоставила пользователям возможность легально разблокировать загрузчик и заменить прошивку на смартфонах на базе Android, серии Xperia (<http://unlockbootloader.sonymobile.com/>). Но в целом процесс получения прав root до сих пор часто напоминает ходьбу по минному полю.

Да кому нужен этот root?

Если вы зайдете на сайт или форум, где упоминается проблема получения прав суперпользователя на Android, то обязательно обнаружите там ЧАВО, где данному вопросу посвящен специальный раздел, в котором написано, что root дает примерно следующее:

- » Полный контроль над всей системой.
- » Возможность изменять системные файлы. В частности, можно изменить следующее:
 - » темы [themes];
 - » приложения, установленные на устройстве по умолчанию (карты, календарь, часы и тому подобное)
 - » загрузочный образ для восстановления после сбоя [recovery image];
 - » загрузчик [bootloader];
 - » Toolbox (программу, поставляемую с Android и позволяющую выполнить ограниченный набор простейших операций, к которым мы привыкли в командной строке Linux, типа ls), например, на Busybox;
 - » собственно загрузочный образ [boot image];
 - » добавить исполняемые двоичные Linux-файлы;
 - » ну и мелочи всякие, вроде информации по часовым поясам.

Да-да, без специальной программы вроде *ClockSync* (спасибо Сергею Баранову) «непереход» на зимнее и летнее времена не будет работать гарантированно, если у вас после того судьбоносного указа не было обновлений. А с учетом того, что даже HTC забывает включить в базу поясов третий по величине город России, то с переходом времени там вообще полный бардак.

- » Возможность запускать приложения, которые требуют более полного контроля над системой, чем пользователю выдается по умолчанию. Примеры таких программ:
 - » SuperUser (позволяет отклонить или подтвердить претензии пользовательской программы на права «супер-юзера»);
 - » Task Manager For Root (возможность прибить процессы и программы, которые обычным способом не убиваются);
 - » а также вирусы и трояны — просто будьте осторожны.
- » Возможность полностью сохранить текущее состояние системы и в случае необходимости восстановить его.
- » Переместить кэш браузера/карт/маркета на microSD-карту.
- » Переместить установленные приложения на microSD-карту.
- » Перегрузить свой телефон с терминала с помощью команды su и reboot.

Если ничего из перечисленного вам не нужно и все в вашем аппарате вас устраивает, то не забивайте голову этой проблемой. Если же у вас чешутся руки что-то поменять, то давайте посмотрим, что вы сможете потерять при использовании root:

» Больше не будет фирменных обновлений OTA [Over-the-air]. В принципе, данную возможность можно не отключать, но в этом случае, даже если все пройдет хорошо, вы потеряете права root, и всю процедуру придется начинать по новой, причем не факт, что она будет той же. Фирменные обновления затыкают известные дырки, которые эксплуатируются при получении прав root, поэтому часто инструкция по получению таковых выглядит следующим образом:

- 1 Проведите «даунгрейд» до версии XXX.
 - 2 Получите права root.
- » Продавцы, да и представители производителей состроят вам козью морду по причине наличия нестандартной прошивки, если вы принесете им аппарат по гарантии. Поэтому, чтобы не облегчать им жизнь, перед походом следует восстановить прежнюю прошивку. Для этого обязательно следует сделать резервную копию непосредственно до прошивки.
 - » Вы теряете ощущение (возможно, совершенно безосновательно), что кто-то, кроме вас, использует ваш телефон.
 - » Вы теряете необходимость искать информацию о том, что такое права root и как их получить, так как вы уже сделали это!
 - » Если ваш телефон был продан с привязкой к оператору, а вы его разблокировали, то не очень разумно обращаться к оператору с какими-либо просьбами относительно вашего аппарата без хотя бы поверхностного «заметания следов».
- Обратите внимание, что получение прав root — это нештатная операция для большинства присутствующих на рынке моделей

телефонов и планшетов на Android. Все, что вы натворите – это целиком ваша самодеятельность, и никто вас утешать не будет.

То есть вы имеете полное право поломать свое устройство любым удобным для вас способом. К сожалению, в некоторых случаях особого выхода и не видно.

В качестве примера можно привести электронные книги, выпускаемые Barnes & Noble. В устройствах Nook сочетаются низкая цена, разумная железная начинка, высокое качество изготовления, жесткая привязка к замечательному книжному магазину, для отоваривания в котором необходима кредитка американского банка, и отсутствие возможности поставить стороннее программное обеспечение для чтения того же fb2. Поэтому в российских реалиях всякий Nook подвергается перепрошивке или значительной модификации, что превращает его во вполне приличный Android-планшет (доступ в магазин Barnes & Noble при этом может сохраниться).

Среди программ, которым нужен root-доступ, можно отметить, например, следующие:

- » CifsManager – монтирование внешних сетей Samba/NFS;
- » TimeZone Fixer – обновление базы часовых поясов;
- » Button Savior – программная эмуляция стандартных железных кнопок Android;
- » Super Manager – комбайн, включающий и файловый менеджер, и бэкап, и клиент ftp/NFS/cifs, и управление приложениями, и многое другое;
- » Font Changer – смена системных шрифтов;
- » SetCPU – разгон процессора;
- » QuickSShd – sshd (удобно для обмена файлами между телефоном и компьютером);
- » AdFree – блокирование рекламы в приложениях (не всем разработчикам это нравится).

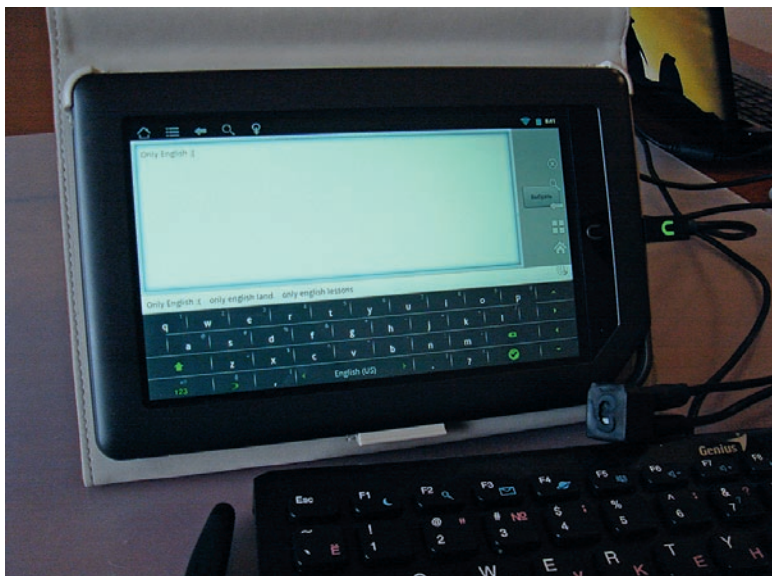
Как получить права root

Предполагается, что получение прав root сопряжено со значительной модификацией прошивки устройства. Здесь следует различать модификации родной прошивки (иногда их называют стоковыми, от stock) и полноценные сторонние сборки из исходных кодов Android, типа CyanogenMod, MIUI или Replicant. Как правило, стоковые прошивки более стабильны, хотя и менее функциональны, нежели полноценные сторонние сборки, но опять же: что именно выбрать, зависит от конкретной модели устройства.

А тут, к сожалению, нет никакого общего и всеобъемлющего рецепта. Правда, работает одно эмпирическое правило: Чем более популярно ваше устройство, тем больше шансов найти подробную инструкцию (включая видео на YouTube) о его программной модификации, и тем больше шансов, что есть прошивки, которые это устройство поддерживают.

Поэтому первое, что следует сделать – это накопить критический объем информации о своем устройстве. Для этого придется потратить несколько часов на специализированных форумах XDA-Developers (<http://forum.xda-developers.com>) или, если чтение на английском вызывает дискомфорт, на 4pda.ru (www.4pda.ru). Необходимо найти ветку или тему, посвященную именно вашей модели, прочитать все, осознать возможные проблемы, запастись картами и устройствами чтения microSD, набраться храбрости и приступить к процедуре, сохраняясь на каждом этапе, чтобы всегда можно было откатиться. Капелька параноии тут будет весьма к месту. Особенно это касается устройств, где имеется дополнительная защита и где, кроме получения root-доступа, необходимо выполнить процедуру S-OFF (Security OFF).

Некоторые действия могут на первый взгляд показаться дикими. Например, для регистрации на Android-маркете Nook



» К Nook Color подключена обычная внешняя USB-клавиатура (нужен либо microUSB – USB, либо USB-переходник типа «мама – мама»). USB-host работает, несмотря на то, что это и не Ice Cream Sandwich. Английская раскладка работает. Для использования русской раскладки нужно что-то, по-видимому, добавить в программной части. Программная клавиатура мешает при наборе (как на снимке). Это лечится установкой Null Keyboard с Google Play.

Touch с новой прошивкой потребует вначале зарегистрироваться на YouTube, и только после этого можно будет устанавливать программы непосредственно с маркета. Временами ощущаешь себя участником какого-то квеста, где результат можно получить, только если выполнишь все действия в строгой последовательности.

К вопросу о сторонних сборках

Для начала – немного информации об официальных сборках. Не будем интересоваться глубокой стариной двухлетней давности. Начнем сразу с Android версии 2.2 Frodo, которая вышла 20 мая 2010 года, последняя корректирующая версия – 2.2.3.

Следующая и последняя версия второй ветки 2.3, Gingerbread, вышла 6 декабря 2010 года. Последняя корректирующая версия – 2.3.7. Именно эта версия в основном стоит на устройствах среднего ценового диапазона, производители которых не забыли своих пользователей и не обделили обновлениями. Именно эту версию вы скорее всего выберете, если задумаете поставить стороннюю прошивку.

Следующая, третья ветка Honeycomb для телефонов не представляет почти никакого интереса, так как здесь Google занялся экспериментами на тему планшетных компьютеров.

Это была своеобразная прелюдия к Android 4.0 Ice Cream Sandwich, который вышел чуть меньше года назад.

Самые топовые телефоны и планшеты используют четвертую версию. Она значительно отличается от второй наличием совершенно естественных возможностей типа USB-хоста (в сторонних прошивках они были в какой-то мере реализованы задолго до) и умением делать снимки экрана «из коробки». Безусловно, изменений гораздо больше, поэтому стан разработчиков сторонних прошивок до сих пор напоминает картину «все уши на фронт», то есть все пият «мороженку с печенькой» с той или иной степенью успеха. Пробовать или не пробовать эту версию – решать вам. Тут до сих пор присутствует значительная степень случайности.

«Первое – следует накопить информацию о своем устройстве.»

» **ClockworkMod Recovery.** Позволяет легко перепрошить устройство, а также забэкапиться перед этим.



» CyanogenMod

CyanogenMod (суаноген – цианистый) является одной из самых популярных и стабильных сторонних сборок. На сегодня гарантированно зафиксировано более 1,7 миллионов установок этой прошивки. На сайте проекта <http://www.cyanogenmod.com/> hostятся более 60 вариантов прошивок для разных типов телефонов и планшетов.

Прошивки разбиты на три категории: стабильные прошивки, кандидаты в релиз и ночные сборки. По идее, стабильные прошивки являются наиболее свободными от ошибок, но так как выявление и исправление ошибок идет непрерывно, то может оказаться, что последняя ночная сборка лучше всего удовлетворяет вашим ожиданиям.

СуаногенMod версии 7 соответствует Android 2.3 Gingerbread. Самая последняя из стабильных версий – 7.2 (2.3.7). СуаногенMod версии 9 соответствует Android 4.0, находится в активной разработке, стабильных версий пока нет.

СуаногенMod постоянно совершенствуется и по своим возможностям превосходит официальные прошивки. Чтобы работать с Android-маркетом, необходимо вместе с прошивкой установить пакет программ Google, код которых закрыт. Планируется, что со временем сообщество СуаногенMod организует свой репозиторий и обявзку вокруг него, но пока приходится использовать не очень-то дружелюбный маркет. Довольно часто СуаногенMod используется в качестве основы для создания специализированных любительскихборок. Например, для упомянутого Nook Color одна из самых популярных и стабильныхборок – Phiremod 7.1.

» MIUI

MIUI (произносится как “Me You I”) разрабатывается китайской фирмой Xiaomi Tech. Часто именно эту сборку устанавливают на китайские андроидофоны. Ее, соответственно, и рекламируют как самую популярную прошивку всех времен и народов (не без влияния численности населения КНР).

В отличие от СуаногенMod, где основной упор идет на базовые программы и код Android, прошивка MIUI часто представляет собой комбайн из уже предустановленных программ. То есть все продумывают уже за вас. На сайте MIUI, <http://en.miui.com/>, можно найти сотни тем и конструктор по сборке комбайна из программ.

У MIUI есть и русскоязычное сообщество, которое обретается на сайте <http://miui.ru> и форумах 4pda.ru.

» Replicant

Несмотря на то, что код Android имеется в свободном доступе, значительная часть системных компонент имеет закрытый код. Специально для «настоящих джедаев свободного программного обеспечения» развивается дистрибутив Replicant (<http://replicant.us/>). В рамках Replicant разработчики поставили задачу заменить все закрытые драйвера и компоненты, дабы окончательно освободить Android.

Меняем прошивки как перчатки

Предположим, что вы установили новую прошивку с root-доступом. Вы все настроили... и вам вдруг показалось, что новая прошивка сделает вас немножечко более счастливым. Тогда, прежде чем начать это делать, необходимо запастись набором инструментов, дабы смена прошивок происходила легко и непринужденно, а в случае каких-либо проблем можно было быстро откатиться назад.

Все три перечисляемые ниже программы требуют для работы прав root.

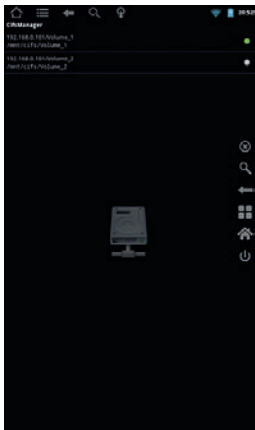
» ROM Manager

ROM Manager написан автором *ClockworkMod Recovery* (<http://www.clockworkmod.com/>) Кушиком Дутта [Koushik Dutta].

ClockworkMod Recovery – это модифицированный загрузочный образ для восстановления после сбоя, который позволяет сделать абсолютно полное резервное копирование системы на microSD, перезаписать новую прошивку и восстановить систему из резерва.

» Так может выглядеть Nook Color после перепрошивки (три экрана, основной экран – средний).

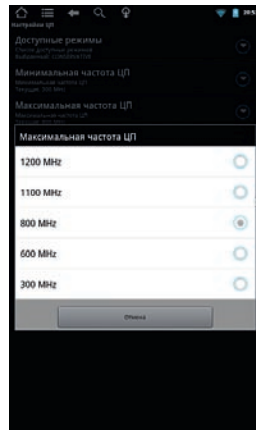




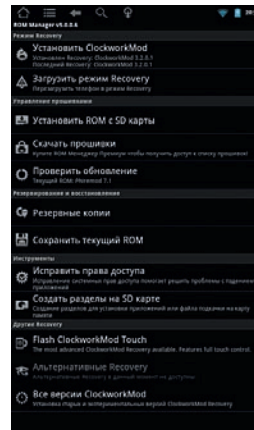
➤ Cifs-manager подключает внешний samba-ресурс.



➤ ClockSync (синхронизация часов с интернет-стандартом).



➤ Исследуем возможности разгона.



➤ ROM Manager — здесь показаны основной экран и экран загрузки прошивок.



ной копии. Имея *ClockworkMod Recovery*, можно смело экспериментировать – при условии, что предварительно забэкапился. Все это работает в специальном режиме, в который нужно перегрузиться. В *SyanogenMod ClockworkMod Recovery* установлен по умолчанию.

ROM Manager – это программа Android, которая позволяет скачать популярные прошивки из интернета и поставить задачу *ClockworkMod Recovery* на предмет бэкапа или перепрошивки. Программа хочет немного денег в обмен на некоторое увеличение функционала. Я думаю, что автору такого чуда имеет смысл подкинуть немного средств.

➤ Titanium Backup Pro

Вам может показаться, что у меня есть признаки легкой паранойи по поводу бэкапов. Если вам так кажется – значит, вы слишком беспечны, потому что я-то нормален.

Больше всего времени при настройке удобного окружения занимает установка и настройка сторонних программ. Это вам не Linux, где все пользовательские настройки хранятся в */home*. Тут каждая программа все таскает с собой, и при ее смерти все настройки сгорают вместе с ней (если автор не побеспокоился о сторонней резервной копии – а он обычно не беспокоится). При перепрошивке все программы, естественно, убиваются, и устанавливаются, а также настраиваются их нужно опять по новой.

Тут на помощь приходит *Titanium Backup Pro*, который позволяет сохранить образы программ и восстановить их по запросу. Программа хочет немного денег – дайте ей их, и в обмен вы получите автоматизацию восстановления и резервное копирование по таймеру.

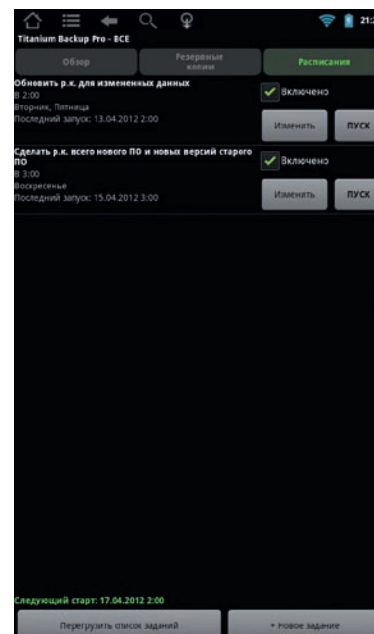
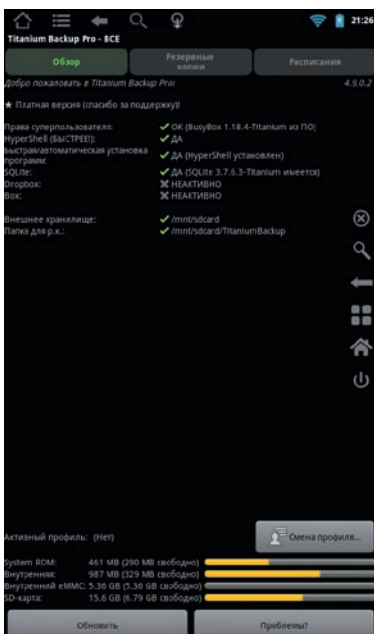
➤ Calibrate battery

После перепрошивки необходимо будет откалибровать батарею. *Calibrate battery* примерно это и делает – точнее, удаляет файл калибровки, а калибровка выполняется уже самой системой. Если этого не сделать, то система может буквально сойти с ума и впасть в непрерывный цикл перезагрузки, пока не разрядится полностью.

Несколько слов после

Далеко пока еще Android'у до удобства Linux. Постоянно приходится искать костыли для выполнения задач, которые, казалось бы, должны решаться на автомате, не обременяя жизнь пользователю.

С другой стороны, заметно, что система постепенно меняется к лучшему, в том числе и за счет свободных наработок. Со временем появится сторонний маркет, и вот тогда можно будет устроить свою личную инфраструктуру с блэкджеком и барышнями. LXF



➤ Titanium Backup Pro (основной экран, список забэкапленных программ, настройки для автоматического бэкапа).



» Как отстоять права root: внедрим теорию в жизнь



Android и Motorola

Александр Хрюкин делится личным опытом обхода тивоизации в телефоне на базе операционной системы Android.



Наш эксперт

Александр Хрюкин

У него есть *gdb*, *Vim* и желание сделать софт свободным.

Тивоизация – это когда вендор занимается привязкой конкретной версии софта к железкам устройства. Подобные ограничения не позволяют модифицировать софт или позволяют это делать только в ограниченных пределах. [Термин «тивоизация» сконструировал Ричард Столлмен из названия компании TiVo, которая применила ПО, лицензированное под GPL, на своих цифровых видеоманитофонах, аппаратно блокировав, однако, возможность модифицировать ПО, – прим. ред.]

Что делать, если вендор выпустил телефон, но не хочет выпускать апдейты к нему?

В прошлом году я купил неплохой такой телефон от компании Motorola с Android под капотом, это был уже мой второй девайс.

Какое-то время меня все устраивало, но Google имеет похвальное свойство выпускать обновления одно за другим и в достаточ-

но короткий срок, и что самое важное – с достаточно большими количественными и качественными изменениями, как в самом фреймворке, так и в ядре Linux.

Мой же телефон был, что называется, тивоизирован по самое не могу, а стало быть, обновления ему никоим образом не светили.

Зачем это нужно конкретно для телефонов со «свободным» Android, мне несколько неясно: на мой взгляд, Motorola пытается катить из года в год одну и ту же телегу, до-

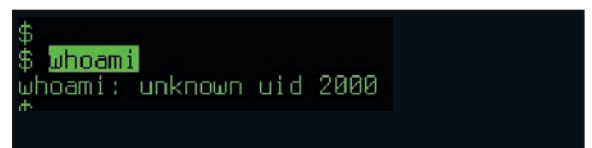
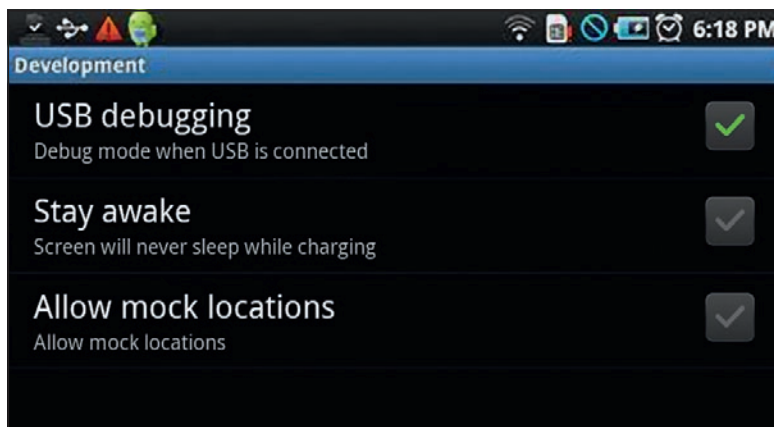
верху набитую подлыми и устаревшими приемами, ради, якобы, «пользовательской безопасности».

Итак, вооружившись старым добрым *Vim*, парочкой компиляторов и *git*, я приступил к ковырянию в программных внутренностях аппарата под названием Motorola Charm MB502.

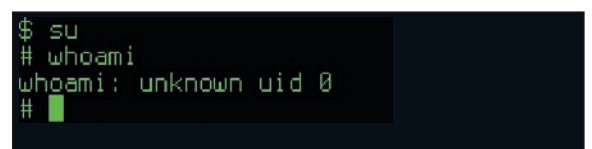
Сначала следовало отделить открытые компоненты от про-

«Google имеет свойство выпускать обновления одно за другим.»

» Включим ADB.



» До руга.



» После этого.



» Осторожно! Идет обновление.

приетарных компонентов или тех GPL-компонентов, для которых компания «забыла» открыть исходные коды.

А если очень хочется поставить свою версию операционной системы, нужно придумать способ, как ее заливать в NAND-память устройства.

Во время разработки инженер может работать с NAND-чипом, например, при помощи JTAG-кабеля, но я подобной роскоши, в силу не зависящих от меня причин (тивоизация же повсюду), не имел. Соответственно, оставался единственный способ: использовать стандартные средства операционной системы, то есть *recovery* + Android Debug Bridge (ADB).

После этого можно подключить телефон USB-кабелем к компьютеру, ввести команду

```
adb shell
```

и попасть наконец-то внутрь устройства.

Чтобы гулять по файловой системе дальше точек монтирования типа */tmp*, */sdcard* и небольшого кусочка */data*, необходимо получить root-доступ.

Детально описывать процесс не буду, скажу лишь, что подобные программы доступны непосредственно в Android-маркете и являются абсолютно бесплатными.

Ну вот, теперь у меня появилась возможность совершать любые непотребства с такими разделами, как */data* (где хранятся личные данные, учетки, пароли и прочее) и */system*, где находится собственно сам Android, то есть его фреймворк, демоны и прочие радости дефолтной мотороловской поставки телефона.

Сейчас я попробую описать, как выглядит процесс замены прошивки устройства у тивоизированного устройства и, скажем, условно открытого.

Например, при переходе с Android 2.1 на 2.3 инженеры Google внесли несколько изменений в ядро Linux, связанных с шифрованием, ну и в 2.3, конечно же, перешли на новую стабильную ветку ядра 2.6.32.

Происходит замена следующим образом: пользователь запускается в режим *recovery* – при помощи мастера обновлений или соответствующей комбинации кнопок на устройстве; дальше *recovery* в соответствии со скриптом может самостоятельно начать процесс обновления, ну или придется руками выбрать пункт "Update".

В случае с тивоизацией же совершенно невозможно прошить что-то еще, кроме апдейта, подписанного ключом вендора, а ключ этот выдают только особо избранным – и простым смертным он недоступен. Взломать его тоже не очень получится: на брутфорс подобных подписей должно уйти время, сравнимое со временем существования Вселенной. Поиск ключа *hash-collision* в теории должен занять меньше времени, но оно все равно значительно дольше человеческой жизни.

Итак, чтобы прошить в NAND-память правильный апдейт, нужен *recovery*, который не проверяет ту заветную цифровую подпись.

Таких *recovery* на самом деле полно: в проекте CyanogenMod используется *ClockWork Recovery*, года два назад были еще несколько, но в итоге все слились в единый проект CyanogenMod.

Процесс компиляции *recovery* в рамках этой статьи я рассматривать не буду, остановимся на этом как-нибудь в другой раз. Сейчас же важен способ запуска нового модифицированного *recovery*, которое не проверяет подписи.

Настало время напомнить, что новое *recovery* записать в Nand невозможно, потому что загрузчик проверяет при старте системы подпись разделом *boot* и, собственно, самого *recovery*.

Значит, нужно запускать с каким-нибудь хитрым костылем. Покопавшись весьма продолжительное время во внутренностях опубликованного Моторолой ядра, его конфига и файловой системы телефона, я обнаружил, что существует двоичный файл под названием *mot_boot_mode*, и, что самое удивительное, его запускает непосредственно ядро.

Исходный код бинарника был закрыт, но, на мое счастье, размер исполняемого файла был всего 6 КБ, а во мне проснулся талант реверс-инженера.

Файл был немедленно дизассемблирован и переписан под свободной лицензией.

Суть программы *mot_boot_mode* сводилась к следующему: при старте системы запускается программа, дальше она читает файл */proc/bootinfo* и спрашивает у него: «А чего ж мне ядро туда понаписало и что же делать?» И в соответствии с содержимым выполняет парочку команд, которые запускают или не запускают Android.

Итак, решено, буду запускать новое *recovery* при помощи этой штуковины.

Примерно за час я дописал еще небольшую часть кода, суть которой сводилась к тому, что при существовании файла */data/recovery_mode mot_boot_mode* остановит загрузку Android, развернет архив с *recovery* из раздела */cdrom*, который я приспособил под хранение нового *recovery*, и собственно запустит его.

Вот подобным нехитрым способом я смог запустить новое *recovery*, которое обладает массой преимуществ по сравнению со стандартным, а именно:

- » возможность сохранения текущего образа телефона на карту памяти;
- » калибровка батареи;
- » установка zip-архива с карты памяти;
- и множество других приятных мелочей.

Весь код легко раздобыть на сайте:

- » <https://github.com/fedya/CharmBootstrap>
- » https://github.com/fedya/motorola_charm_hijack
- » https://github.com/fedya/mot_boot_mode **LXF**

» Все оказалось просто.





» Ваши программные наработки не пропадут даром

ARM и Android: Программирование

Часть 3: Андрей Боровский считывает данные из видеобуфера Android, углубляясь в механизм взаимодействия приложений Linux и системы Android.



Наш
эксперт

Андрей Боровский
В 14 лет сломал школьную локальную сеть. И это оказалось только началом.

Научившись считывать данные из видеобуфера Android и выводить графику на экран устройства, вы сами сможете написать программу, которую на Google Play продают за деньги. Для этого в наше блюдо, приготовленное на C, придется добавить щепотку Java.

Стоит отметить, что от релиза к релизу в составе средств разработки для Android появляется все больше возможностей взаимодействия с системой, предназначенных для C и C++. Вероятно, разработчики Android уже поняли, что они несколько погорячились, оставив возможность разработки для своей платформы только на Java. Уже сейчас под Android можно написать полноценное приложение на C и C++, не содержащее ни одной строчки кода на Java. Правда, код Java все равно будет добавлен в это приложение (компоновщиком) для того, чтобы обеспечить его взаимодействие с остальными компонентами системы. В будущем, возможно, мы увидим интерфейс программирования, предназначенный исключительно для языков, компилируемых в машинные коды.

Программирование для Android при помощи стандартных средств взаимодействия C++ и Java подробно описано в документации и сопровождается примерами, так что останавливаться на нем мы не будем. Вместо этого мы рассмотрим некоторые альтернативные методы и средства разработки, которые хотя и не являются частью Android API, но, тем не менее, позволяют решать многие задачи, возникающие перед программистами.

Как я и обещал в прошлый раз, мы сосредоточимся на работе с графикой.

На рисунке показана приблизительная схема взаимодействия компонентов графической системы Android между собой. Эту схему можно разделить на три уровня. На самом верхнем уровне располагаются приложения Android, написанные на Java (возможно, с использованием библиотек, написанных на C и C++ и вызываемых с помощью JNI). Ниже на схеме показан интерфейс, предназначенный для организации взаимодействия графической системы Java и ядра Linux. Этот интерфейс состоит из нескольких компонентов, каждый из которых реализован в собственной библиотеке, написанной на C или C++. Данные библиотеки являются

обычными разделяемыми библиотеками Linux, и наши программы могут получить прямой доступ к любой из них, хотя пользы от этого будет, вероятно, немного.

Обратите внимание на компоненты SurfaceFlinger и PixelFlinger. По какой-то причине слово «flinger» полюбилось разработчикам Android, и они широко его используют. На русский язык точнее всего было бы перевести эти названия как «жонглер поверхностями» (имеются в виду виртуальные поверхности, на которые приложения выводят графику) и «жонглер пикселями». Помимо

прочего, компоненты промежуточного слоя выполняют программную эмуляцию отсутствующих компонентов графического оборудования. Например, PixelFlinger может эмулировать поддержку трех-

мерной графики, если в системе отсутствует аппаратная поддержка 3D. К этому же уровню я отнес файлы-устройства, предоставляющие непосредственный доступ к видеобуферу системы, хотя, возможно, их следовало бы поместить уровнем ниже.

В соответствии с показанной трехуровневой системой, у разработчика приложений на C и C++ есть три пути взаимодействия с графическими компонентами Android.

Первый путь заключается в том, чтобы использовать код, написанный на C и C++ внутри приложения Java, играющего роль обертки. Этот путь хорош тем, что приложение ведет себя по пра-

«В блюдо, приготовленное на C, придется добавить щепотку Java.»

Как пройти в библиотеку?

Если вы планируете подключать к приложению напрямую такие библиотеки, как libflinger.so или libsurfaceflinger.so, то имейте в виду, что скомпоновать приложение с этими библиотеками на этапе сборки не удастся. Android NDK просто ничего о них не знает.

Единственный подходящий способ – воспользоваться методом явной загрузки библиотек во время выполнения программы, с помощью функций `dlopen()`, `dlsym()` и `dlclose()`. При этом вам все равно понадобятся заголовочные файлы соответствующих библиотек. Вы найдете их в составе NDK, в разделе исходных текстов Android.

вилам, установленным системой, а значит, в процессе его работы мы можем избежать неприятных сюрпризов.

Второй путь заключается в получении доступа к библиотекам, реализующим интерфейс между Java и ядром системы. Как уже отмечалось, эти библиотеки вполне доступны нашим программам. Недостаток данного подхода заключается в том, что программы, взаимодействующие с нижними уровнями абстракции, не учитывают события, происходящие на более высоких уровнях, что может привести к конфликтам и непредвиденному поведению этих программ. И это не говоря о том, что сам промежуточный уровень, не являющийся частью официального API Android, может быть подвержен неожиданным и плохо документированным изменениям.

Третий путь заключается в непосредственном доступе к устройству видеобуфера. Для решения некоторых задач этот способ может быть приемлем, но в общем случае он страдает тем же недостатком, что и способ, описанный выше: программа, обращающаяся к буферу, неизбежно конфликтует с другими программами, которые работают с системой на высоком уровне.

Из перечисленных выше трех подходов к работе с графикой мы рассмотрим два наиболее рациональных: прямое взаимодействие с видеобуфером и использование тонкой оболочки Java вокруг кода программы, написанного на C. Заодно мы рассмотрим другой, более простой способ написания Make-файлов для сборки приложений Linux под Android. В отличие от рассмотренного в предыдущей статье, этот способ скрывает от нас многие интересные детали и работает только с NDK, зато позволяет создавать более компактные Make-файлы.

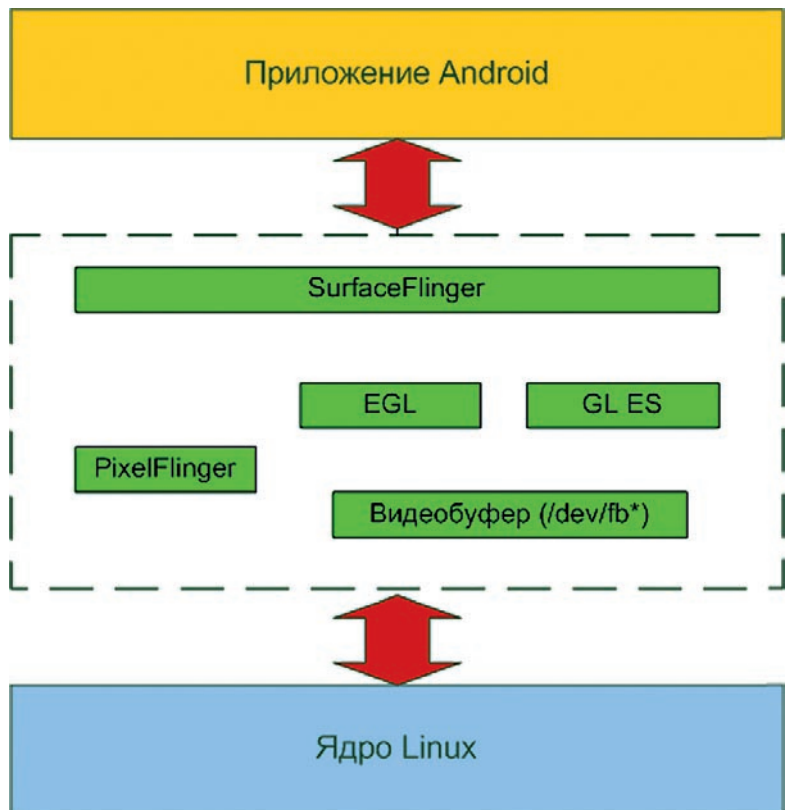
Считывание данных из видеобуфера

Устройство видеобуфера `/dev/graphics/fb0` позволяет получить прямой доступ к тому, что происходит на экране устройства Android. Данные, которые можно считать из файла `/dev/graphics/fb0`, представляют собой поток значений пикселей, где первым идет верхний левый пиксель, а последним – нижний правый. К этому надо добавить, что, поскольку устройства Android используют двойную буферизацию, из файла `/dev/graphics/fb0` можно прочитать содержимое сразу двух кадров видеобуфера. Чтобы преобразовать этот поток «сырой» информации в наглядный графический формат, необходимо знать по крайней мере два параметра видеосистемы устройства: формат хранения пикселей, который тесно связан с количеством цветов, отображаемых устройством, и разрешение экрана устройства по вертикали и по горизонтали. Пиксели на экранах Android могут быть представлены 16-ю, 24-мя и 32-мя битами. Количество пикселей на экране колеблется от 240×320 до 800×1280 .

Рассмотрим фрагмент программы `fboutput`, которая делает снимок экрана устройства и сохраняет его на диске в формате BMP. Весь текст программы вы найдете на прилагающемся диске, в то время как фрагмент, представленный ниже, определяет параметры видеобуфера.

```
int fd;
fb_fix_screeninfo fi;
fb_var_screeninfo vi;
fd = open("/dev/graphics/fb0", O_RDWR);
ioctl(fd, FBIOGET_FSCREENINFO, &fi);
ioctl(fd, FBIOGET_VSCREENINFO, &vi);
```

Мы открываем файл `/dev/graphics/fb0` как обычный файл Linux, а затем делаем два вызова `ioctl()`. Первый вызов позволит получить фиксированные характеристики экрана, второй – переменные характеристики. Структуры, в которые вызовы записывают данные (переменные `fi` и `vi` соответственно), имеют тип `fb_fix_screeninfo` и `fb_var_screeninfo`. Оба типа объявлены в заголовочном файле `/include/linux/fb.h`. Мы не будем описывать все



поля этих структур; скажем только, что после выполнения приведенных выше вызовов `ioctl()` переменная `vi.xres` будет содержать число пиксельных столбцов в данном режиме работы экрана устройства, `vi.yres` будет содержать число строк, а переменная `vi.bits_per_pixel` – число битов на один пиксель. Переменная `fi.smem_len` будет содержать размер видеобуфера в байтах (в данном случае – объем памяти, необходимый для хранения двух кадров буфера).

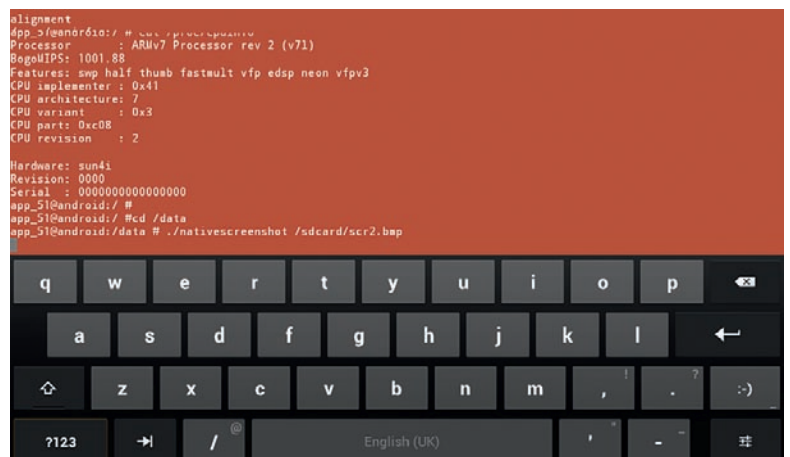
Самый простой и быстрый способ получить доступ к данным видеобуфера заключается в том, чтобы отобразить память видеобуфера в память нашей программы:

```
void * bits = mmap(0, fi.smem_len, PROT_READ | PROT_WRITE,
MAP_SHARED, fd, 0);
```

В результате указатель `bits` указывает на начало области данных буфера. Объем памяти, занимаемый одним кадром, можно посчитать по формуле

$$vi.xres * vi.yres * (vi.bits_per_pixel \ll 3).$$

» Схема взаимодействия приложения Android и графических компонентов.



» Снимок экрана Android, сделанный с помощью программы `fboutput`.

Для получения скриншота нам осталось считать данные одного кадра из буфера и конвертировать их в формат BMP. Все эти операции проделывает программа *fboutput*, и останавливаться на них подробно мы не будем.

Система позволяет нам не только считывать, но и записывать данные в видеобуфер, что должно соответствовать выводу пикселей на экран, однако на практике ничего путного из этого не выйдет, так как наша система вывода будет конфликтовать с выводом других программ.

Мы могли бы собрать программу *fboutput*, используя в качестве шаблона Make-файл, введенный в предыдущей статье, но поступим иначе – воспользуемся системой сборки программ NDK/SDK, которая также пригодится нам в следующем примере. Прежде всего, стандартная система сборки Android предполагает, что собираемые файлы расположены в определенных директориях.

В случае программы *fboutput*, которая все же не является стандартным приложением Android, структура проекта выглядит довольно просто. В директории проекта расположена поддиректория *jni* (она называется так потому, что обычно содержит код, к которому остальные части приложения обращаются помощью одноименного механизма), где содержится файл исходных текстов нашей программы, а также два файла сценариев сборки – **Android.mk** и **Application.mk**. Содержимое файла **Application.mk** выглядит не просто, а очень просто:

```
APP_PLATFORM := android-9
NDK_APP_OUT := ./
APP_ABI := armeabi
```

Нетрудно догадаться, что в этом файле мы указываем версию API (в нашем случае – android-9), корневую директорию для результатов сборки и формат двоичного кода, который мы хотим получить (armeabi).

Файл **Android.mk** несколько сложнее:

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := fbout
LOCAL_SRC_FILES := capturescr.c
include $(BUILD_EXECUTABLE)
```

Переменная `LOCAL_MODULE` содержит имя, которое будет присвоено двоичному файлу, полученному в результате сборки. Переменной `LOCAL_SRC_FILES` следует присвоить список файлов

исходных текстов нашей программы. Отдельного внимания заслуживает последняя строчка, которая определяет, чем именно должен быть результат сборки. Поскольку обычно код C и C++ используется в приложениях Android в форме разделяемых библиотек, в большинстве случаев эта строка будет выглядеть как `include $(BUILD_SHARED_LIBRARY)`

но мы хотим получить на выходе файл самостоятельной программы и поэтому используем константу `BUILD_EXECUTABLE`.

Все, что нам теперь осталось сделать – перейти в корневую директорию нашего проекта и вызвать в ней скрипт *ndk-build*, входящий в состав NDK. Если сборка прошла успешно, то в поддиректории корневой директории проекта **libs/armeabi** появится

исполняемый файл нашей программы. Не удивляйтесь, что файл появился в директории **libs**: все-таки основное предназначение NDK – сборка разделяемых библиотек. Тем не менее, NDK знает о том, что

собирает исполняемый файл программы. В процессе сборки программа автоматически компонуется с фрагментом кода, содержащим функцию `_start()`, которая вызывает нашу функцию `main()`. Скомпилированную программу нужно загрузить в устройство Android (или его эмулятор) и в окне консоли Android командовать нечто наподобие

```
fboutput screenshot.bmp.
```

В результате будет создан файл **screenshot.bmp**, содержащий снимок экрана устройства.

«Нашим программам доступны все средства графики Android.»

Работа с графикой с помощью SDL

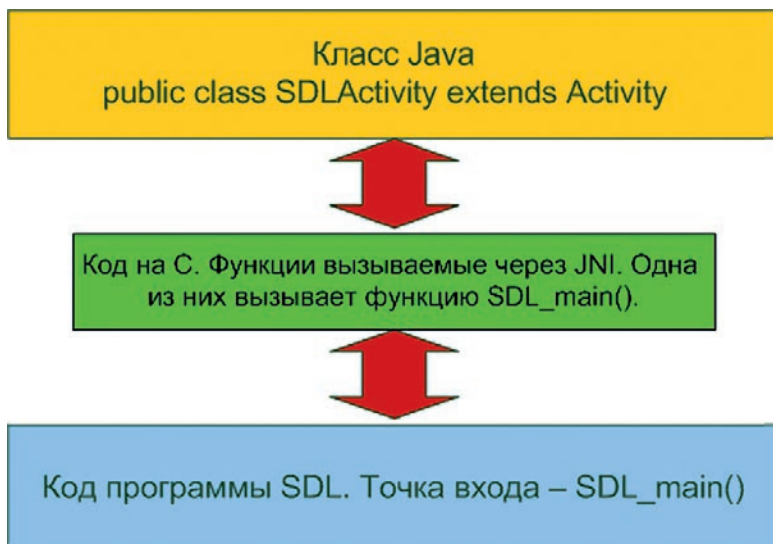
Как мы уже видели, нашим программам доступны все средства вывода графики Android, начиная с OpenGL ES и заканчивая видеобуфером. Осталось только встроить графический вывод нашего приложения в общую механику работы операционной системы. Самый простой способ сделать это – написать обертку

Android и многозадачность

Хотя Android – многозадачная система, эта многозадачность работает не совсем так, как в ОС для ПК. Предполагается, что в каждый момент времени пользователь работает только с одним приложением (окно которого занимает весь экран устройства). Другие процессы, окна которых не видны, рассматриваются системой как процессы заднего плана (возможны еще процессы, у которых вообще нет окон, но мы на них останавливаться не будем). Процессы заднего плана могут продолжать работу, но в случае, если системе не будет хватать памяти, любой из этих процессов может быть автоматически завершен и выгружен. При этом система не только не спрашивает пользователя о том, согласен ли он на выгрузку приложения, но и не предупреждает его об этом (сама программа, естественно, получает от системы соответствующие сигналы).

В идеале принудительная выгрузка и повторный запуск приложения вообще должны быть незаметны для пользователя. В момент выгрузки приложение должно сохранять информацию о своем состоянии и восстанавливать это состояние при повторной загрузке. На практике так происходит не всегда. При этом, разумеется, совершенно недопустима ситуация, когда принудительная выгрузка программы приводит к каким-то ошибкам. В этом аспекте программирование для Android отличается от программирования для ПК, где предполагается, что программа будет работать до тех пор, пока пользователь не захочет завершить ее (и в этом случае ответственность за сохранение данных ложится на пользователя).

➤ Схема взаимодействия кода Java и SDL в приложении Android.

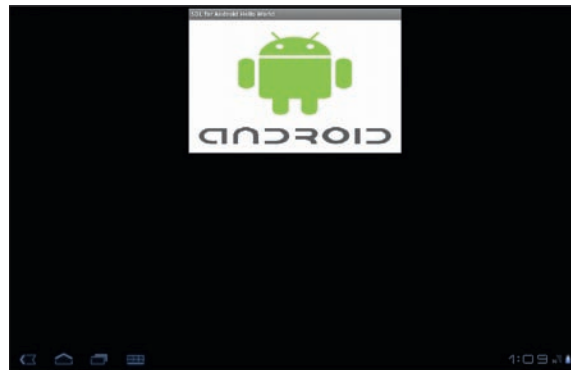


на Java, которая будет вызывать полезный код нашей программы с помощью JNI. Поскольку одна из задач, которую мы ставим перед собой, заключается в упрощении переноса на Android программ, предназначенных для настольных версий Linux, возникает вопрос, насколько должен измениться код программы, написанной на C или C++. Ответ на этот вопрос зависит от того, насколько переносимая программа по своему поведению должна быть похожа на стандартное приложение Android. В самом простом случае код программы, написанной на C или C++, не придется менять вообще. Но полученная в результате программа будет вести себя не совсем так, как обычные приложения Android. Более глубокая «андроидизация» программ потребует внесения изменений в их архитектуру в соответствии с требованиями ОС Android.

В качестве примера программы, работающей с графикой, которую можно с минимальными изменениями перенести на платформу Android, мы рассмотрим простую программу, использующую библиотеку SDL (www.libsdl.org). Выбор SDL не случаен. Эта библиотека уже давно используется программистами-умельцами в процессе переноса игр на мобильные платформы (достаточно назвать игру *OpenTTD*, в которую можно поиграть и на платформе Android). Кроме того, начиная с новейшей версии SDL – SDL 2, платформа Android поддерживается библиотекой официально. Итак, допустим, мы хотим портировать на Android простую программу, написанную с использованием SDL:

```
#include "SDL.h"
int SDL_main(int argc, char *argv[]) {
    SDL_Window *win = NULL;
    SDL_Renderer *renderer = NULL;
    SDL_Texture *bitmapTex = NULL;
    SDL_Surface *bitmapSurface = NULL;
    SDL_Rect rect;
    int running = 1;
    SDL_Init(SDL_INIT_VIDEO);
    win = SDL_CreateWindow("SDL for Android Hello World", SDL_WINDOWPOS_CENTERED, SDL_WINDOWPOS_CENTERED, 800, 480, SDL_WINDOW_SHOWN);
    renderer = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED);
    bitmapSurface = SDL_LoadBMP("data/android.bmp");
    bitmapTex = SDL_CreateTextureFromSurface(renderer, bitmapSurface);
    SDL_FreeSurface(bitmapSurface);
    while (1) {
        SDL_Event e;
        if (SDL_PollEvent(&e)) {
            if (e.type == SDL_QUIT) {
                break;
            }
        }
        SDL_RenderClear(renderer);
        SDL_RenderCopy(renderer, bitmapTex, NULL, NULL);
        SDL_RenderPresent(renderer);
    }
    SDL_DestroyTexture(bitmapTex);
    SDL_DestroyRenderer(renderer);
    SDL_DestroyWindow(win);
    return 0;
}
```

Программа целиком позаимствована из документации SDL. Единственное, что она делает – выводит на экран картинку, загруженную из файла **data/android.bmp**. Поскольку на этот раз результатом сборки станет программа с расширением APK, нам придется выполнить все формальности, связанные с созданием



► Приложение SDL на экране Android.

проекта приложения Android: организовать структуру директорий, как того требует система сборки Android, и добавить необходимые файлы, описывающие проект. Все это вы найдете на диске (проект называется **sdl-android**). Все файлы находятся на своих местах, за исключением исходных текстов библиотеки SDL. Я не стал включать их потому, что SDL 2.0 все еще пребывает в состоянии разработки и к моменту выхода журнала поддержка Android в этой библиотеке, скорее всего, будет расширена и улучшена.

Итак, рассмотрим бегло структуру проекта приложения *SDL-Android*. Исходный текст самой программы находится в директории **jni/src**. В директории **jni/SDL** нужно разместить исходные тексты библиотеки SDL, произведя в них небольшие изменения (описанные в файле **README** для Android, входящем в состав дистрибутива библиотеки). В директории **src** находится файл исходных текстов на языке Java. Этот файл реализует ту самую «обертку», о которой говорилось выше. В директории **assets/data** находится файл BMP, из которого программа загружает изображение. Во время выполнения собранной программы этот файл будет загружаться из системы ресурсов приложения.

Приступим к сборке. Сначала, с помощью знакомого нам скрипта *ndk-build*, мы соберем ту часть кода, которая написана на C. В результате у нас появятся две разделяемые библиотеки: **libSDL2.so** и **libmain.so**. Вторая библиотека содержит код нашей программы, то есть функцию **SDL_main()**.

Обратите, кстати, внимание на новые директивы в файле **jni/src/Android.mk**:

```
LOCAL_SHARED_LIBRARIES := SDL2
LOCAL_LDLIBS := -IGLESv1_CM -llog
```

Первая директива указывает на то, что библиотеку **libmain.so** следует связать с библиотекой **libSDL2.so**, собранной в рамках этого же проекта. Вторая директива указывает на необходимость подключения библиотек **libGLESv1_CM.so** и **liblog.so**, которые являются частью Android.

Теперь нам придется воспользоваться инструментом, которым мы прежде не пользовались – программой *ant*. Эта программа предназначена для сборки проектов Java; она же создаст нам файл с расширением APK. В корневой директории проекта командою

```
ant -debug
или
ant -release
```

И получаем файл APK, который можно установить на устройство Android так же, как и любое другое приложение.

Написание программ для Android на C требуется не только хакерам и тем, кто хочет перенести на мобильную платформу написанный ранее код. Только используя C и C++, мы можем сделать то, что часто требуется в высокопроизводительных программах: обратиться напрямую к возможностям процессора ARM. Об этом – в следующей части. **LXF**

Сравнение

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

Хостинг проектов

Где уютнее всего пристроить вашу дивную новую программу Linux?
Майк Сондерс выступает арбитром смотра-конкурса лидеров...



Про наш тест...

Мы создали проекты у каждого из пяти провайдеров хостинга, оценивая, насколько просто было их настроить и разобраться с интерфейсом. Мы прошли стандартный процесс администрирования проекта — индивидуальной настройки web-страниц, загрузки файлов, настройки каналов коммуникации и активации системы отслеживания ошибок. Попутно мы примерили на себя роль администратора проекта, жаждущего разместить нашу работу в Интернет и привлечь новых участников. Дружелюбие к пользователю интерфейса администратора и простота внесения изменений были основным фактором нашей оценки. Другими факторами, повлиявшими на нашу оценку, были производительность провайдера (мы проверяли их в разные дни), навязчивость рекламы и легкость нахождения проекта среди других постельцев этого провайдера.

Наша подборка

- » BerliOS
- » GitHub
- » Google Code
- » Savannah
- » SourceForge

Вас осенила блестящая идея проекта свободного ПО. Вы написали код, истребили ошибки, разработали броские логотипы и готовы одарить мир своими результатами. Единственная проблема — где это все поместить? Наскоро сляпать web-страницу и закинуть ее в сеть может каждый, но современные пользователи ожидают большего. Им подавай интерактивные сайты, где можно поместить сообщение

на форум и где есть системы отслеживания ошибок. Они рассчитывают на wiki, чтобы читать и обновлять документацию по программе, и на систему контроля версий, чтобы быть всегда на переднем крае вашего кода. Если вы хотите завлечь в свой проект других разработчиков, предусмотрите разные варианты участия. К счастью, возможностей для размещения вашей работы в сети немеряно. Но, как и при выборе дистрибутива, это раз-

нообразие может поначалу сбить с толку. Поэтому в нашем Сравнении мы возьмем пять самых популярных сервисов хостинга проектов и рассмотрим, что они предлагают, насколько просты в использовании и для каких проектов подходят. Требования небольшого любительского клиента IRC, например, существенно отличаются от требований к ядру операционной системы, так что мы будем оценивать еще и гибкость этих сервисов.

Настройка и web-хостинг

Насколько просто заставить ваш проект работать?

Легко ли создать проект? Google Code позволяет создать сайт с минимумом действий – всего лишь укажите имя, краткое описание и лицензию. После завершения процесса вы получите простоватый с виду сайт со ссылками на обновление статуса, скачивание и wiki. В плане индивидуальной настройки особо не размахнешься, и большинство проектов Google Code выглядят как из одного стручка, но есть и свои плюсы: навигация по этим сайтам действительно простая, и на главной странице посетители видят рейтинг активности, позволяющий судить о посещаемости и популярности проекта. URL проекта выглядит следующим образом: <http://code.google.com/p/myproject>.

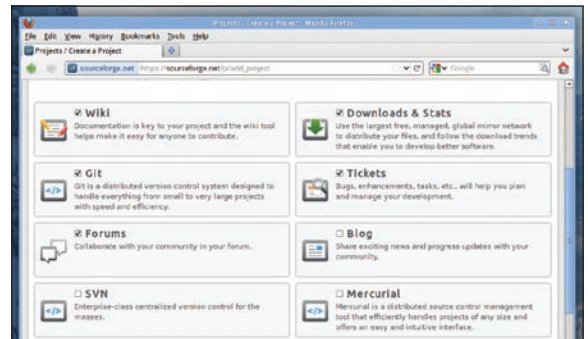
Создать сайт в BerliOS будет посложнее: придется прочесть изрядное количество текста и пройти через юридические соглашения. Но главная помеха – необходимость ждать (обычно 24 часа) разрешения от человека на начало работы над проектом. Это, конечно, уменьшает на BerliOS свалки ненужного хлама и число отдающих фантастикой заявок «я та-акое напишу!», но все же является ограничивающим фактором. Получив разрешение, загружайте на сайт HTML и изображения через SCP, но на вашей главной странице

должен быть логотип BerliOS со ссылкой на главный сайт. URL имеют вид <http://my-project.berlios.de>.

SourceForge предлагает завлекательный процесс подписки, который сразу же демонстрирует имеющиеся функции – wiki, блог, форумы, SVN и прочее – в сопровождении изображений. Можно отметить галочкой нужные опции, и хотя обдумывать все это при подписке отчасти и преждевременно, нам нравится такое информирование на ранней стадии. За один щелчок вы окажетесь на странице Administration (о ней мы поговорим позже). Затем продолжайте, загружая HTML и выбранные изображения. Все это взаправду быстро, а в использовании – проще не бывает.

GitHub предоставляет бесплатный хостинг для проектов с открытым кодом (ценовые предложения зависят от желаемого количества репозитория) – вам нужно только указать имя своего проекта, и репозиторий *Git* создастся автоматически. Соломки непосвященным здесь никто не подстилает – после регистрации вы просто получите список команд *Git* для ввода. Основное внимание web-интерфейса отдается просмотру репозитория, но можно также загружать страницы HTML.

Savannah нас помучила, отказавшись принимать наш пароль (11-символьная



➤ Добавить в проект функции в SourceForge очень легко.

комбинация заглавных и строчных букв и цифр ей не угодила) и заставив расшифровать контрольное изображение. Требования к хосту строгие: в процессе регистрации просят подтвердить наличие у всех ваших исходных файлов действующих уведомлений об авторских правах, наличие копии лицензий в tar-архивах, и т.п. Нужно также предоставить ссылку на tar-архив, чтобы администраторы Savannah могли проверить, все ли в порядке; и документация гласит, что в описании следует говорить «GNU/Linux», а не просто «Linux», если вы хотите получить разрешение сразу. После этого загружайте HTML и получайте сайт на www.nongnu.org/myproject.

Вердикт

SourceForge
★★★★★
BerliOS
★★★★★
GitHub
★★★★★
Google Code
★★★★★
Savannah
★★★★★
➤ SourceForge стартовал с отрывом.

Отслеживание ошибок

Хорошо ли извещается о проблемах и их приоритетности?

ВerliOS предоставляет систему сообщения об ошибках на базе web, с категориями, приоритетами и группами; и администраторы могут отмечать ошибки как открытые, закрытые (решенные), дубликаты или недействи-

тельные. Можно даже создавать настраиваемые запросы SQL для просмотра ошибок заданного типа. Интерфейс системы Tickets в SourceForge более привлекательный, но в ней не хватает возможности присвоения ошибкам приоритетов, что может стать проблемой, если вам приходится перелопачивать сотни сообщений об ошибках.

Система Issues в Google Code также довольно проста, но она любезно предлагает текстовый шаблон для отчета, в котором пользователю предлагается описать все

его шаги по воспроизведению проблемы. Хорошо бы у всех хостинг-провайдеров было нечто подобное, чтобы уменьшить количество абсолютно бесполезных отчетов, содержащих одну эпохальную строку: «Ваша программа работает некорректно».

Система GitHub, как ни странно, весьма примитивна и просто дает возможность присвоить ошибки определенным разработчикам и установить метки – то есть даты, когда определенные ошибки должны быть исправлены. Savannah тоже отличается незатейливостью и не балует пользователя избытком информации.



➤ Google Code требует массу подробностей.

«Можно создавать запросы SQL для просмотра ошибок заданного типа.»

Вердикт

BerliOS
★★★★★
Google Code
★★★★★
SourceForge
★★★★★
GitHub
★★★★★
Savannah
★★★★★
➤ BerliOS наиболее всесторонний.

Администрирование

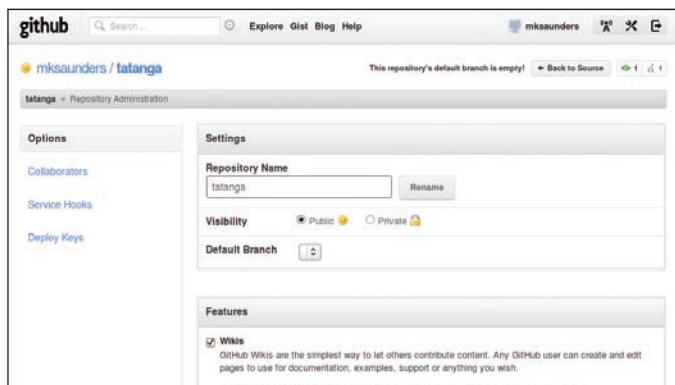
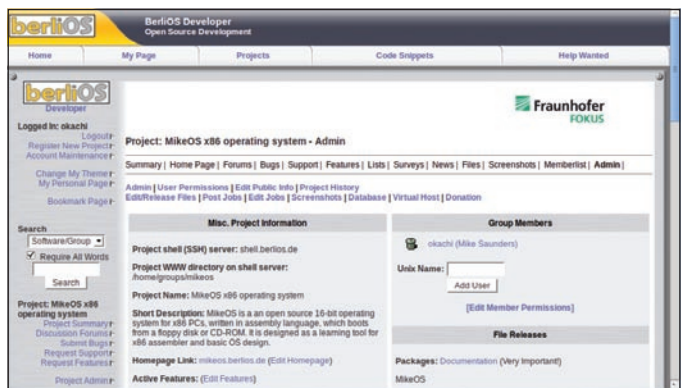
Насколько прост и эффективен в использовании интерфейс?

Как главный разработчик своего проекта, львиную долю своего времени вы будете проводить в web-интерфейсе Administration: загружать новые релизы, добавлять скриншоты, отвечать на отчеты об ошибках или разрешать споры на форумах. Здесь красота функций не является: неважно, сколько в интерфейсе забавных CSS-эффектов, вам просто надо сделать свое дело за минимум щелчков мыши (или нажатий клавиш,

фанаты Ratpoison). Мы оценили удобство и эффективность отдельных функций, а здесь сконцентрируемся на общем впечатлении от интерфейса: простоте навигации, скорости отклика и наличии раздражающих факторов. Отметим, что у многих наших провайдеров есть инструменты для автоматизации предоставления релизов: например, *Shipper* (<http://www.catb.org/~esr/shipper/>) позволяет создавать новые релизы в SourceForge из командной строки.

BerliOS ★★★★★

Хотя интерфейс BerliOS насыщен текстом, и порой от него рябит в глазах, разобраться с ним не так уж сложно. Доступ к основным функциям осуществляется через список в верхней части, и для большинства функций имеется очевидная ссылка Admin, дающая доступ к настройке. Однако наше главное нарекание на BerliOS – то, что здесь не сохраняется статус нахождения в системе для многократных браузерных сессий. Иначе говоря, при каждом закрытии и повторном открытии браузера для получения доступа к функциям администратора нужно снова входить в систему. Это не всемирная катастрофа, потому что большинство современных браузеров позволяют сохранять данные, но добавляет лишнее действие каждый раз, когда вы решите запустить Firefox и поработать с последними отчетами об ошибках. Однако в целом это достойная попытка.



GitHub ★★★★★

Для GitHub репозиторий – это все, и web-интерфейс Dashboard в основном нацелен на взаимодействие с ним. Пользователи могут просматривать дерево последнего исходного кода, видеть, кто вносит изменения, и «наблюдать» за обновлением проекта, когда кто-то загружает новый код. Можно также просматривать ответвления кода и узнать, подойдут ли заплатки к основному дереву. Очень помогают полезные экранные снимки и руководства, содержащие команды и объясняющие, как управлять репозиториями и устранять проблемы; они разделены по категориям для разного уровня пользователей (Начинающий – Beginner, Средний – Intermediate и Продвинутый – Advanced). Есть даже примитивный проект “Hello World”, его можно применять для тестирования. В целом, интерфейс хороший и профессиональный, и делает исходный код приоритетом – хороший дизайн для совместных проектов.

Сбор статистических данных

Кто просматривает ваш сайт и что с него скачивают?

Возможность отслеживать, кто просматривает ваш сайт, и проводить мониторинг деятельности в разделе Downloads обеспечит вашему проекту отличную обратную связь. По умолчанию Google Code поможет разве что в предоставлении индивидуальной статистики по скачиванию; чтобы добыть более подробную информацию, вам придется соединить ваш проект с учетной записью Google Analytics – лишней и довольно занудный шаг. GitHub работает намного лучше, предоставляя детальные

графики, объясняющие все в таких понятиях, как ipract (представляющий объем внесенного кода) и runchcard (загрузки по времени и дате). SourceForge тоже снабжен привлекательными графиками, отображающими активность с точки зрения посещения сайта, форумов, отслеживания проблем и системы контроля версий, и для них можно указать диапазон дат. Он также предоставляет добавочную информацию о том, где находятся посетители, о самой популярной ОС, используемой для просмотра страницы вашего про-

екта, и т.д. Для BerliOS ситуация несколько сложнее, поскольку есть отдельный вид для мониторинга статистики сайта (добавьте “usage” в конец вашей основной страницы) и еще отдельная страница статистики в разделе Admin. Все это сопровождается подробными и научнообразными графиками, но столь тонкого контроля, как в SourceForge, все-таки нет. А Savannah предоставляет отчеты по ошибкам и скачиваниям в виде гистограмм, и отдельный инструмент (очень мощный AWStats) по доступу на web-страницу.

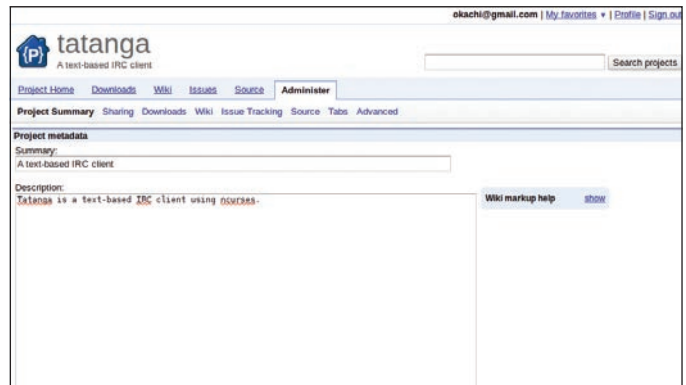
Вердикт

- GitHub ★★★★★
- SourceForge ★★★★★
- BerliOS ★★★★★
- Savannah ★★★★★
- Google Code ★★★★★

» GitHub и SourceForge просты в использовании.

Google Code ★★☆☆☆☆

В обычной манере Google, все опции объединены в единственной вкладке Administer, а та подразделяется на подкатегории для проекта, скачиваний, совместных действий (например, для других участников), wiki, отслеживанию проблем и прочего. Почти все опции предоставлены в текстовых окнах, так что легкость доступа здесь не такая, как в интерфейсе SourceForge с флажками, но у опытных разработчиков не будет проблем с навигацией. Ограниченная степень индивидуальной настройки интерфейса доступна через панель Tabs, позволяющую предоставлять альтернативные страницы для разных разделов, таких, как Downloads и исходный код (очень удобно, если вы планируете разместить их на другом сайте). Это среднестатистический интерфейс, и он устроит опытных разработчиков, но производит такое впечатление, словно он делался в спешке и местами не завершен.



Savannah ★★★★★★

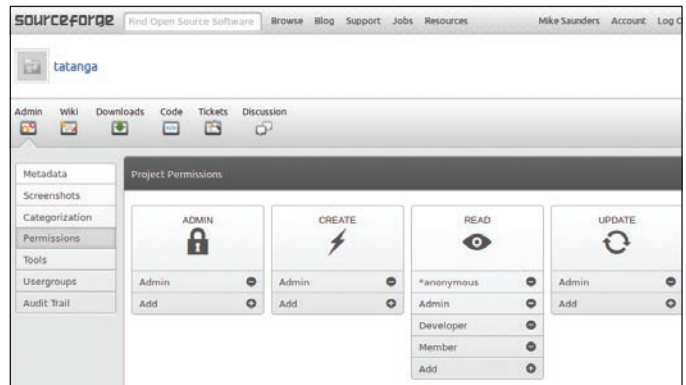
Интерфейс Savannah понятен и привлекателен, отличается жизнерадостной любительской атмосферой и GNU-подобностью. Здесь есть напоминание о том, что пожертвования помогают сайту выжить, но во всем остальном он выглядит более ориентированным на сообщество, чем остальные, с цветными значками и знаменитым символом GNU, который оживляет пейзаж.

Списки рассылки, отчеты об ошибках и задачи управляются через панель Administration; здесь же добавляются и удаляются пользователи. Включение и отключение функций производится за несколько щелчков мыши, и при желании разместить некоторые функции на стороне можно задать альтернативные URL. Решив поучаствовать в других проектах, вы можете индивидуально настроить свой профиль, перечислив свои умения и опыт – другие разработчики, ищущие таланты, смогут найти вас и связаться с вами.



SourceForge ★★★★★★

И, наконец, страницы администрирования SourceForge всех привлекательнее внешне: они пестрят значками и яркими описаниями. Все аккуратно распределено по категориям с тщательным и точным контролем разрешений в проекте (например, у кого еще есть права администратора, кому разрешено делать обновления кода, и т.д.). На стадии настройки есть удобный экран, где можно активировать новые функции, такие как wiki или база данных MySQL, одним щелчком. Правда, периодически возникают странности: в нашем тестовом проекте мы получили сообщение, что «не было создано значков», хотя и способа загрузить значки не предлагалось. Не будучи идеален, данный интерфейс – самый всесторонний и простой в использовании из всех интерфейсов, которые мы опробовали в нашем Сравнении, и очень удобен для тех, кто берется за администрирование впервые.



Контроль версий

Управление изменениями.

В некоторых проектах, особенно с малым количеством участников, использование системы контроля версий выглядит перебором. Но это хороший способ отслеживать даже минимальные изменения в древе кода и в экстренном случае вернуться к рабочей версии.

По умолчанию SourceForge активирует доступ Git для проектов, но через панель Admin можно также активировать Mercurial и Subversion, если вы предпочитаете их. Очень удобно, что web-интерфейс позволяет генерировать команды

оболочки для доступа к репозиториям Git по различным протоколам (например, SSH и HTTPS), и внесенные коды можно просматривать через сайт. BerliOS на шаг обогнал соперников, предлагая поддержку CVS в дополнение к трем имеющимся в SourceForge – она активируется флажками в разделе Administration на сайте. Нас особенно порадовало, что BerliOS предлагает разные способы просмотра древа исходного кода через сеть – например, ViewCV и WebSVN для репозитория Subversion и GitWeb и CGit для репозито-

риев Git. Если у вас возникли трудности с одним интерфейсом, можете переключиться на другой.

В GitHub просто есть Git, но с привлекательным web-интерфейсом для просмотра репозитория (включая файлы исходного кода с выделением синтаксиса). В Google Code тоже Git, плюс Mercurial и Subversion в качестве опций и незатейливый web-браузер кода; а Savannah предлагает все три эти опции вместе с GNU Arch (но вам придется порыться по документам, чтобы его настроить).

Вердикт

- BerliOS ★★★★★★
- Savannah ★★★★★★
- GitHub ★★★★★★
- Google Code ★★★★★★
- SourceForge ★★★★★★

» В Savannah есть выбор, но побеждает BerliOS.

Коммуникации

Возможность общения с пользователями.

Выбор метода коммуникации для вашего проекта бывает непростой задачей. Многие администраторы предпочитают сперва создать форумы, и это может показаться наилучшим решением, поскольку они доступны через web-интерфейс. Однако проблемы здесь – спам и вероятность того, что пользователи начнут тратить больше времени на придумывание себе подписи, чем на общение. Списки рассылки зачастую лучше, потому что ваши сотрудники по разработке могут сами выбрать свой клиент электронной почты для организации общения.

SourceForge включает модуль форума, на уровне среднего: с подфорумами, базо-

вой разметкой для создания таблиц и ссылок и подписки электронной почты для мониторинга определенных потоков. Однако навигация по форумам и размещение сообщений кажутся какими-то дергаными, и все усыпано атрибутами SourceForge, ужимающими сам форум до небольшой части страницы. Не то чтобы это было вовсе непригодно для использования, но мы предпочитаем подход phpBB, когда форум занимает центральную часть, а знаки SourceForge временно исчезают. Вне форумов можно также активировать списки рассылки с помощью *Mailman*.

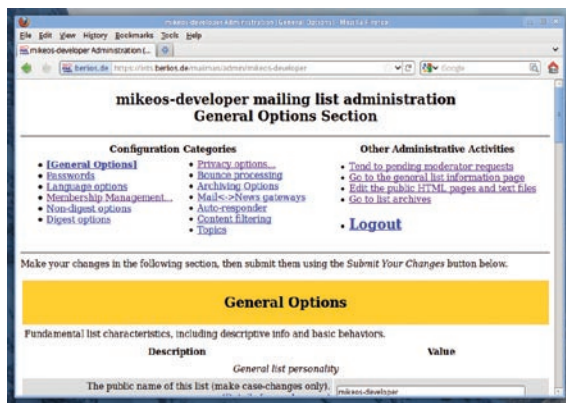
BerliOS немногим лучше: он заявляет вам прямо в интерфейсе администрирования, что «если вы добавите форум, он не может быть изменен или удален!». Ну почему?! К вящей путанице, раздел Forums на странице Project по умолчанию пуст, но, перейдя в панель Admin, вы увидите три созданных форума – Open Discussion, Help и Developers [Открытое обсуждение, Помощь и Разра-

«SourceForge включает модуль форума на уровне среднего.»

ботчики]. Исследовав интерфейс далее, вы обнаружите, что все они помечены как «удаленные», хотя и преспокойно продолжают существовать. Это несколько разочаровывает в BerliOS, но зато здесь отличные списки рассылки благодаря *Mailman*.

Как и следует ожидать от Google, коммуникационное устройство в Code перекинуто на другой сервис, а именно Google Groups. Проблем с Groups у нас не возникло, но это как-никак дополнительный набор шагов по настройке. Впрочем, web-интерфейс для просмотра и управления постами неплох.

В Savannah нет форумов, но списки рассылки доступны через *Mailman*. GitHub не предоставляет форумов или списков рассылки, и нам пришлось выдать ему ноль очков. Это может показаться жестоким, и сторонники GitHub будут протестовать, восклицая, что весь сайт посвящен *Git* и мы обязаны понимать, что наличие форумов только прибавило бы головной боли. Однако простая система списков рассылки пригодилась бы администраторам, желающим разместить свой проект на GitHub и обойтись без внешних сервисов.



» *Mailman* – на редкость надежный и гибко настраиваемый менеджер рассылок.

Вердикт

- SourceForge ★★★★★
- BerliOS ★★★★★
- Google Code ★★★★★
- Savannah ★★★★★
- GitHub ★★★★★

» Списки рассылки у большинства есть, но форумы слабоваты.

Дополнительные функции

Менеджеры задач, wiki, пересмотры и подпроекты.

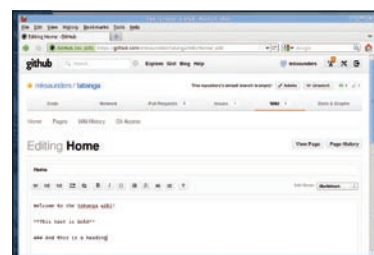
Savannah предоставляет менеджер задач, отчасти похожий на ее систему отслеживания ошибок, но в нем разработчики могут размещать цели и отмечать сроки этапов проекта, указывая приоритеты и процент завершенности работ. Есть также менеджер новостей для обновлений, сопровождаемый лентой RSS. GitHub предоставляет отличную возможность для wiki, отлаженную в стиле GitHub – с различными форматами разметки и кнопками для применения разных стилей, если вы не знакомы с языком разметки.

Google Code также предоставляет wiki, но во многих аспектах сервиса производит такое впечатление, словно вы используете скрипт CGI конца девяностых, а не со-

временное web-приложение. Работу свою он делает, но те, кто занялся администрированием впервые, вряд ли почувствуют себя здесь комфортно. В остальном Google Code не предлагает ничего примечательного – как мы уже упоминали, от вас ждут обращения к другим сервисам Google для добавления функциональности.

В SourceForge есть несколько удобных дополнительных функций, таких как обзоры, где конечные пользователи могут писать краткие рецензии на ваш проект. Имеется также простая wiki, урезанная до синтаксиса Markdown; пользоваться ею легко, но недостает гибкости реализации GitHub. Полезной функцией SourceForge является возможность создания подпроектов.

BerliOS применяет MediaWiki для своей системы wiki, и она исключительно мощная (не зря ее выбрали в Wikipedia), но в интерфейсе BerliOS выглядит инородной. Существует также менеджер пожертвований.



» С интерфейсом wiki GitHub редактирование превращается в приятную прогулку по парку.

Вердикт

- BerliOS ★★★★★
- SourceForge ★★★★★
- GitHub ★★★★★
- Savannah ★★★★★
- Google Code ★★★★★

» BerliOS и SourceForge оснащены лучше других.

Хостинг проектов

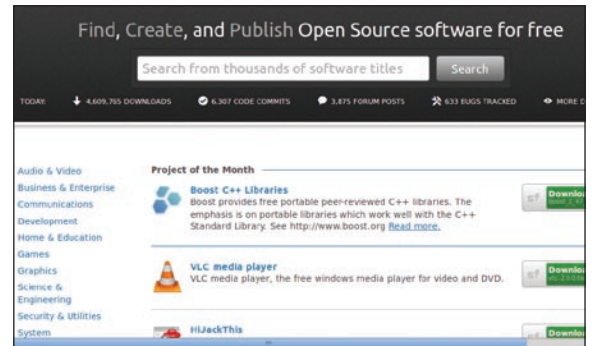
Вердикт

У нас два явных лидера: SourceForge и BerliOS. Первый пользуется репутацией свалки недоделанных проектов – хостинг-провайдера проектов, где 98 % проектов никогда не выходят из стадии планирования. Пару лет назад автор перевел свой проект операционной системы с SourceForge на BerliOS, потому что первый стал жутко тормозить и кишеть ошибками, причиной чему могли стать десятки тысяч незаконченных проектов, сжирающих ресурсы. Однако сегодня SourceForge отличается удивительной быстротой, простотой для начинающих и множеством функций, превратившись во вполне достойный выбор.

BerliOS с очень незначительным отрывом занимает вторую позицию, предлагая хороший набор функций для продвинутых пользователей в меньшем объеме настроек. SourceForge похож на столицу, где кипит жизнь и где трудно пробиться, а BerliOS напоминает приятный областной город, где есть все удобства вдали от шума и всяких

сложностей. Поскольку проекты сначала должны получить разрешение от человека, мы бы сказали, что качество программ в BerliOS выше, чем в SourceForge, потому что чересчур эксцентричные идеи отсеиваются (и там большой простор для поиска названия – на SourceForge почти все хорошие названия уже разобраны заброшенными проектами).

Однако BerliOS чреват проблемой: мы не знаем, долго ли он еще протянет. BerliOS был создан Институтом Фраунгофера [Fraunhofer Institute] в Германии как ресурс для проектов с открытым кодом, и отлично работал 12 лет, но, к сожалению, лишился финансовой поддержки. В конце 2011 года BerliOS разослал сообщения администраторам проектов, уведомляя, что он близок к закрытию. К счастью, в последний момент его спасла группа добровольцев, но на его долгожительство нет никаких гарантий. Для мелких проектов это не особо важно,



но если кому нужна прочная web-база, им, возможно, не стоит сейчас связываться с BerliOS.

И, наконец, чтобы умерить лавину гневных сообщений от любителей GitHub, скажем, что он отлично выполняет свою работу по предоставлению сервисов Git. И это – его основная цель; если вам нужен только Git, это выбор на пять звезд. Но в качестве провайдера ему не хватает ключевых функций.

» За годы существования у SourceForge были взлеты и падения, и на наш вкус, у него слишком много заброшенных проектов, но зато наилучший баланс между функциональностью и удобством использования.

«SourceForge похож на столицу, а BerliOS напоминает приятный областной город.»

I**SourceForge** ★★★★★Сайт: www.sourceforge.net

» Доступный и полнофункциональный – отец всех хостинг-провайдеров для проектов.

IV**GitHub** ★★★★★Сайт: www.github.com

» Блестяще отлажен для любителей Git, а в остальном функциями не блещет.

II**BerliOS** ★★★★★Сайт: www.berlios.de

» Второе место с мизерным отрывом, хотя есть сомнения в его выживаемости.

V**Google Code** ★★★★★Сайт: <http://code.google.com/projecthosting>

» Нежданное разочарование в большом G: просто голый костяк инструментов.

III**Savannah** ★★★★★Сайт: <http://savannah.gnu.org>

» Местами не хватает отделки, но неплох для желающих отбиться бесплатно.

Обратная связьПопробуйте один из этих хостинг-провайдеров и расскажите нам о своем опыте: lx.f.letters@futurenet.co.uk.**Рассмотрите также...**

В Сети есть масса других хостингов; большинство из них специализированные. Интересно, что провайдеры нередко применяют исходный код от других – например, движок SourceForge является ответвлением GNU Project, которое стало Savane и на котором сейчас работает Savannah. Gna (www.gna.org) – еще одно предложение с акцентом на свободном ПО. Размещение вашего проекта у одного из этих провайдеров – знак вашей верности делу свобод-

ного ПО. Если вы фанат Debian, попробуйте Alioth (<http://alioth.debian.org>); любителям *Git*, ищущим альтернативу GitHub, стоит обратить внимание на Gitorious (www.gitorious.org); даже Европейский Союз не остался в стороне со своим Open Source Observatory and Repository (www.osor.eu) – попыткой привлечь общественные организации на континенте к использованию программ с открытым кодом. **ЛХФ**

WIKIOTICS



Свободное ПО помогает людям делать мир лучше. И вот двое таких людей: Нейл Салливан [Neil Sullivan] и Лоран Савайет [Laurent Savaete]. Они — создатели Wikiotics, и их миссия — облегчить всем изучение иностранных языков. Мы заловили их на FOSDEM, чтобы разузнать об их мотивах, об убыточности образования и о коварных венгерских разговорниках.

LXF: Итак, Wikiotics — это полностью открытый, редактируемый ресурс. А он для учителей или для обычных людей?

Нейл Салливан: Для всех. Это wiki, поэтому кто угодно в сообществе может его редактировать. Создавался он в том числе и в образовательных целях, но вот я — не учитель, а все-таки планировал уроки. Возможность самостоятельно организовать свое обучение и подобрать подходящие материалы для себя или для друга — вот для чего изначально предназначена эта система.

LXF: Какие, по-вашему, имеет преимущества wiki перед более привычными сайтами для изучения языков по методу «сверху вниз»?

НС: Методика вообще вещь субъективная. У каждого учителя свои методики. Когда я изучал китайский, каждый раз, когда мы добирались до очередного модуля в учебнике, мы начинали с вычеркивания из нового словаря всех их дефиниций и вписывания тех, что говорил преподаватель, слегка подправленных.

Возвращаясь к методике — учителя очень ценят возможность редактировать материал. И когда мы пересекаемся с профессиональными методистами на конференциях, они все время отмечают, что преподаватели языка терпеть не могут все построенное по принципу «сверху вниз», потому что: «Да кто вообще так учит; это не под нашу программу; теперь вот придется перестраивать подачу материала и критерии успеваемости».



LXF: Понятно, а кто второй энтузиаст, тот, что из Калифорнии — Джим?

НС: Мы с Джимом вместе работали, и когда я стал поговаривать, что современные обучающие программы плохи, и как было бы легко создать новые, а он ответил: «Да запросто», и начал их создавать. Он физик, сейчас учится в аспирантуре в Калифорнии. И у него большой опыт в сфере свободного ПО, хотя бы по сравнению со мной. Я скорее занимаюсь этим в социальном, политическом ключе, а он как раз программист, он-то и выстроил всю инфраструктуру. Два месяца назад мы все вместе провели неделю в Нью-Йорке, сводя воедино весь наш опыт организации занятий, когда мы пробовали различные системы обучения на аудитории, чтобы комбинировать их в уроках.

Мы полностью перекроили интерфейс сайта и прорисовали все заново на бумаге, рассуждая: «Вот вы попали на главную страницу, и что дальше? Как указать людям простой и доступный путь к обучению или преподаванию?»

LXF: На какой стадии находится проект сейчас? Он живой?

НС: Безусловно. Его пользователями пока являются люди, с которыми мы работаем индивидуально, это всего лишь горстка. Но, опять-таки, много времени у нас ушло непосредственно на разработку самого ресурса. Теперь же, имея обновленный интерфейс, которым мы активно занимаемся, и разные типы уроков, мы абсолютно уверены в его жизнеспособности. Можно составить программу обучения или отдельные занятия, оригинальные и для учителей, и для учеников различных уровней. Например, у нас уже есть уроки, основанные на выборе картинок. Сайт такой можно сделать и самому, но мы исходим из того, что учителя — люди занятые, и им некогда на досуге подучить CSS и HTML, набросать сайтик и поискать хостинг.

Лоран Савайет: Я бы еще раз хотел подчеркнуть, что мы не просто подходим к обучению с позиции учителя. Мы движемся к построению эффективной модели взаимообучения, где вы, к примеру, являетесь носителем английского языка, желающим выучить итальянский, и ищите итальянца, который хочет изучить английский; это вполне реально.

НС: Да, и говоря о построении уроков, мы составляем уроки с целью наполнить систему содержанием: ведь гораздо проще выбрать что-то подходящее из готового материала, нежели начинать все с нуля.

АС: Плюс — одно дело, если речь идет о Нью-Йорке или Париже, но если вы живете в маленьком городе и хотите выучить вьетнамский, найти там носителя языка может быть совсем не просто.

LXF: Если уроки может редактировать, кто угодно, как вы удержите троллей, которые могут превратить все в хаос?

НС: Знаете, давным-давно в шоу Монти Пайтона [Monty Python] был такой скетч — «Кривой венгерский разговорник», где Джон Клиз [John Cleese] идет в табачную лавку и при помощи, как он думает, простейших фраз из разговорника пытается купить сигареты и коробок спичек. Но, в стиле Монти Пайтон, предложение «коробок спичек, пожалуйста» там переводится как



«у меня полный катер угрей», и в итоге ситуация превращается в полный фарс. И это проблема почти всех обучающих языковых проектов, в которые вовлечено много людей. Если вы член сообщества и знаете всех остальных участников, вы в общем имеете понятие о благонадежности предоставляемых ими материалов. Но человек, приходящий на сайт, знать не знает, кто автор материалов.

Мы с самого начала закладывали на такую возможность и нашли решение в трех зонах расположения контента в wiki: во-первых, есть большая открытая wiki, в которой редактировать и добавлять информацию может любой. Вы создаете учетную запись, входите под своим именем, а дальше – делаете что хотите и прилагаете совместные усилия, чтобы система оставалась такой же гибкой, открытой и доступной. Помимо этого, для зарегистрированных пользователей есть еще две зоны. Есть личная зона – то есть, введя на wikiotics учетную запись и имя, вы попадаете в свой раздел и создаете страницы, которые разрешено редактировать только вам. А посетители сайта будут видеть, что у этого пользователя есть коллекция уроков и профиль, на котором сказано: «Учитель из Антверпена».

И они, конечно, могут решить, что вы шутник или спамер, но если в вашей пользовательской зоне действительно спам или приколы, то кроме вас их там разместить было некому, и доверие к остальным вашим материалам, соответственно, пропадет. Тот же принцип защиты информа-

ции и в групповых зонах, только для пользователей, затевающих совместную работу по причине учебы в одном кружке или на факультете. Кроме того, это еще и способ создавать учебные коллекции и готовые архивы.

К примеру, вы просмотрели материалы от разных учителей английского, и многие из них неплохи, но это не совсем то, что вам нужно. И вы можете выбрать те, что вам понравились, скомпоновать их и подправить «под себя».

Или вы хотите, чтобы аудио звучало гармонично – тогда поперебирайте разные коллекции уроков и найдите материалы со сходными голосами. Или можете взять и заменить все картинки в наглядных уроках, чтобы там были одни собаки, и лексику сделать другую. А раз вы любитель собак, то и счет, и предлоги, и цвета, и размеры, и что угодно осваивайте по ним, безо всяких дурацких картинок с людьми, домами и кошками.

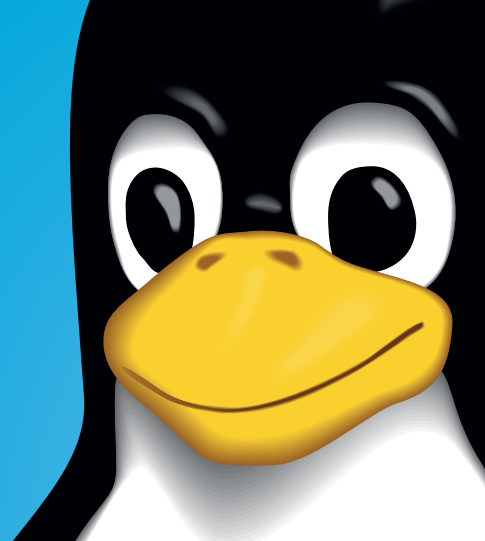
LXF: Выходит, тут есть и рекламная подоплека. Коль скоро вами внесен достойный вклад, ваша репутация не пострадает, если кто-то влез и нагадил.

НС: Вот это и нужно преподавателям языка. Ведь им трудно порой почувствовать отдачу от своей работы или признание, трудно найти учеников. Как оценить, кто хороший педагог? Студенты – клиентура наивная: они не могут сравнить вашу подачу материала с чьей-либо еще, если не успеют поучиться у вас обоих. **LXF**

О ВЗАИМООБУЧЕНИИ

«Мы составляем уроки с целью наполнить систему содержанием.»

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:

- 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  center

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

Linux — лидер

Ваша ОС с открытым кодом — лучшая: она смело идет вперед, а остальные только увязываются следом. Маянк Шарма вам об этом напомнит.

Скажите кому-нибудь, что Linux — лучшая ОС, и клеветник в ответ приведет вам неточные и устаревшие данные по количеству пользователей.

Но ваша ОС не только лучшая: она еще и научила всяким трюкам своих проприетарных собратьев.

Будь у Linux еще и фамилия, то отчество для нее выводилось бы от инновации — ген инновации пронизывает всю экосистему Linux. Linux была не первой ОС, но за годы своего существования восприняла и отточила классические метафоры компьютерных технологий и даже создала свои, которые затем были подхвачены другими ОС.

В нашей статье мы рассмотрим вклад, внесенный Linux в современные компьютерные технологии, на рабочий стол, в серверную и даже при встраивании в различные устройства. Возможно, не все инновации, предлагаемые рынком сообществом Linux, нам по душе, и их авторы могут многому еще поучиться у других; но не стоит винить их за энтузиазм.

Сообщество

Наверное, величайшей инновацией сообщества Linux является само сообщество. Стиль разработки сообществом существовал и до Linux, но он скромно отсиживался на заднем плане, поскольку на повестке дня стояли стиль разработки за закрытыми дверями, предлагаемый проприетарным ПО, да и само проприетарное ПО. А потом Линус Торвальдс [Linus Torvalds], отправив свое знаменитое сообщение в список рассылки, навсегда изменил правила создания про-

шинство серверов в Интернете и превеликое множество систем управления контентом.

Одной из первых инноваций сообщества Linux, весьма полезной для пользователей, стал live CD. Хотя первые live CD Linux появились в начале 1993 года, подлинную популярность у пользователей завоевал дистрибутив Knoppix, основанный на Debian.

Теперь такие инструменты, как *Remastersys* и скрипты *Linux Live*, позволяют каждому создать собственный live CD. Сейчас многие популярные дистрибутивы распространяются в виде устанавливаемых live CD, а многие другие имеют официально поддерживаемый вариант Live CD.

Помимо демонстрации и установки дистрибутива Linux на жесткий диск, live CD Linux завоевал популярность как инструмент восстановления и отладки системы, а также для более специализированных задач, например, настройки высокопроизводительного вычислительного кластера.

«Потом Линус Торвальдс, отправив свое сообщение, навсегда изменил правила.»

грамм, исправления ошибок и оказания помощи пользователям.

Помимо самого Linux, разработка сообществом дала жизнь таким программам, как web-сервер *Apache* и СУБД *MySQL*, на которых работает боль-

Рабочий стол

Ваш рабочий стол — это вершина айсберга инноваций Linux. Хотя все варианты среды рабочего стола за долгие годы прошли долгий и трудный процесс эволюции. Началось все с заимствования концепций других рабочих столов, чтобы предоставить пользователям внешне знакомую среду, а потом переходить к улучшению их удобства.

Темп развития инновационного рабочего стола Linux был настолько высок, что некоторые из предлагаемых им концепций только сейчас начали применяться в других системах.

Поиск по рабочему столу

Пока Windows занимался украшением своих заточенных инструментов поиска цветными анимированными символами (помните щенка Ровера?), инструменты поиска по рабочему столу Linux осваивали новые хитрости: например, контекстный поиск по архивам электронной почты, и многое другое.

Поиск в настольной системе всегда был сильной стороной Linux. Такие инструменты, как *grep*, *find* и *awk*, намного опережали время. По мере того, как файлы на рабочем столе становились все сложнее и обзаводились метаданными, появились такие графические инструменты, как *Beagle*, *Recoll* и прочие, позволив вам просматривать гигабайты данных на дисках за доли секунды.

Теперь у нас есть такие инструменты, как *Nepomuk*, *Strigi* и *Zeitgeist*, которые не просто регистрируют файлы пользователя, но и отмечают действия пользователя, например, посещаемые сайты, и ведут протокол их общения в IM. Они глубоко интегрированы в соответствующие рабочие столы — KDE 4, Gnome 3 и Ubuntu Unity, и повышают удобство и функциональность. А если вы полагаете, что виджеты и апплеты — это исключительно территория Apple, подумайте еще раз. За эти годы

Компьютер для нетерпеливых

Спустя годы разговоров о компьютерах мгновенного действия, где-то в 2008-м на прилавках стали появляться устройства, загружаемые менее чем за 20 секунд. Эти компьютеры дают пользователям доступ к столь важным функциям, как выход в Интернет, обмен быстрыми сообщениями и VoIP, причем значительно снижая время загрузки.

Практически любой из компьютеров мгновенного действия работает на Splashtop OS или на ее ребрендовой или измененной версии.

Под покровом Splashtop OS работают Linux и различные программы с открытым кодом. Можете считать ее ускоренным дистрибутивом Linux.

Изначально эта ОС существовала только в виде предустановленной среды вместе с постоянной операционной системой на поддерживаемых устройствах. Однако сейчас ее можно установить изнутри Windows на любом устройстве. После установки пользователи смогут загрузить или Windows, или среду Splashtop.

рабочий стол Linux получил множество движков для виджетов, таких, как *gdesklets*, *adesklets*, *screenlets* и *SuperKaramba*.

«Android ОС на базе Linux поддерживает виджеты с начала 2009 года.»

Позднее *SuperKaramba* была интегрирована в KDE 3.5 как часть пакета *kdeutils*, но продолжала оказывать влияние на рабочий стол KDE 4, Plasma Desktop. Отображение в виде «приборной доски [dashboard]» в Plasma заменило собой старую функцию Show Desktop. Включение dashboard скрывает все окна и выводит на первый план виджеты.

В KDE 4 виджеты под названием «плазмиды [Plasmoid]» охватывают все, от прогноза погоды на рабочем столе до значков, меню приложений и панелей. Среди их целей было облегчение пользователям задачи изготовления собственных плаз-

мидов. В отличие от Apple Dashboard Widgets или Microsoft Gadgets в Windows 7, плазмидов намного больше, и они выполняют самые разнообразные задачи. Один из наших фаворитов среди плазмидов — Lancelot, для запуска приложений. Вне рабочего стола виджеты также существуют — и служат тем же целям на смартфонах. Android OS на базе Linux поддерживает виджеты с начала 2009 года.

Умные панели

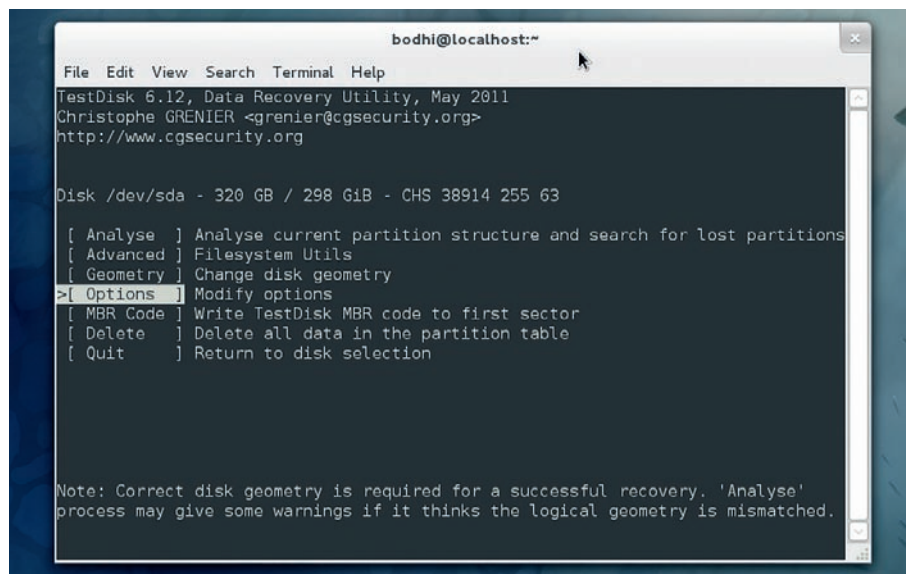
Рабочие столы Linux не изобрели панель задач, лоток или док, но наша реализация куда удобнее и имеет больше возможностей настройки, чем проприетарный оригинал.

Панель задач в KDE превосходно настраивается. Вы можете перемещать ее по всему экрану, а также перемещать элементы внутри нее. Элементы можно вытаскивать за пределы панели задач и добавлять сколько угодно новых панелей задач. Начиная с KDE 4, панель задач реализуется как виджет Plasma.

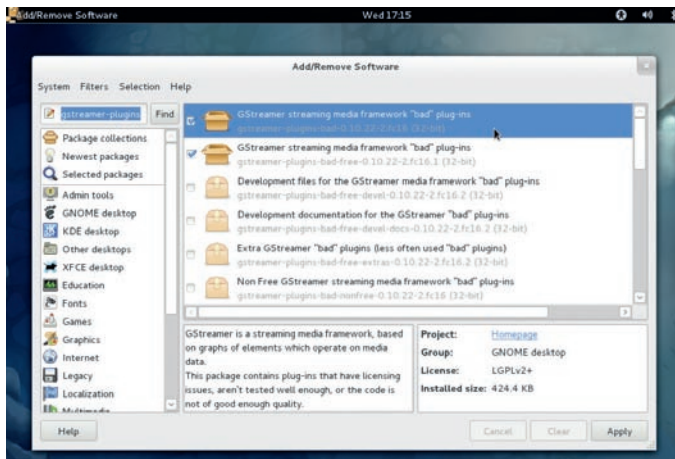
Ubuntu Unity покончил с традиционными панелями задач и панелями, перейдя на программу запуска приложений под названием *Dash*, заменившую собой выпадающее меню классического Gnome. Инновация *Dash* — в предоставлении поиска по приложениям, настройкам, файлам и даже музыке из одного местоположения. А что самое лучшее, она даже может показать вам, какие приложения вы можете загрузить из Ubuntu Software Center. Еще одна инновация — линзы, расположенные в нижней части *Dash*, которые можно использовать для поиска по файлам или музыке.

И Unity, и Gnome 3 теперь оба используют док. Инновация здесь заключается в интеграции значков Unity Dock в приложения. Вы можете щелкнуть правой кнопкой по значку Thunderbird dock, чтобы создать сообщение или просмотреть адресную книгу.

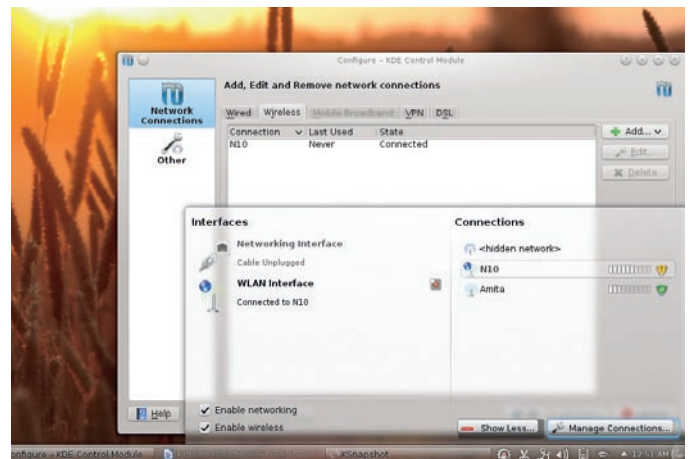
Еще одной инновационной функцией рабочего стола Linux являются уведомления рабочего стола. Например, в Gnome Shell они ненавязчиво



➤ Благодаря ncurses, командная строка Linux обзавелась графическим интерфейсом.



➤ Linux инкорпировал родные 64-битные модули расширения Flash и Java до того, как его примеру последовал Windows.



➤ Инструменты, подобные Network Manager, прикрывают дружелюбным к пользователю графическим интерфейсом сложные задачи.

появляются внизу экрана. Затем имеется Unity MeMenu, которая гарантирует, что вы не пропустите ни быстрых сообщений, ни писем электронной почты, ни событий в вашей социальной сети.

Интеграция с облаком

Дистрибутивы Linux, в частности, Ubuntu, обладали способностью синхронизировать данные с Cloud задолго до iCloud. Сервис облачного хранения Ubuntu, именуемый Ubuntu One, дебютировал в начале 2009 года и был включен в релиз Ubuntu 9.04.

Помимо резервного копирования и синхронизации файлов из вашей установки Ubuntu, у сервиса также есть клиенты для Android, с такими расширенными функциями, как возможность немедленной загрузки фотографий, сделанных на устройстве Android, в облако Ubuntu One. На рабочем столе, помимо отдельных файлов, сервис может синхронизировать контакты из заметок Thunderbird и Tomboy.

Хотя будущее Mandriva туманно, последний раз мы слышали, что там работают над созданием сервиса Mandriva Sync. В дополнение к синхронизации файлов и папок, Mandriva хотела оснастить

свой сервис таким образом, чтобы он мог синхронизировать данные из приложений, такие, как контакты в адресной книге и закладки в браузере.

OpenSUSE также работает над интеграцией в дистрибутив сервиса облачного хранения с открытым кодом ownCloud DIY. Инструмент Mirall создан специально, чтобы помочь начинающим пользователям настроить собственное облако и создать директорию загрузки для зеркального отображения данных.

Сравните с Windows, который представит свой сервис облачного хранения SkyDrive несколько позднее в этом году, в Windows 8.

Виртуальные рабочие столы

Темп инноваций на рабочем столе Linux вырос за последние годы, и нет недостатка в уникальных инновационных функциях.

«Будучи давним пользователем рабочего стола Linux, я нахожу, что таких простых функций, как виртуальный рабочий стол, по-прежнему нет в качестве родной функции в Windows, и это только начинает осваиваться в Mac OS X. Виртуальный рабочий стол резко повышает производительность, и эта тенденция теперь вполне видна, с но-

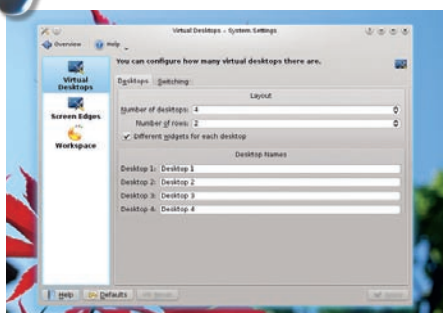
вым поколением «изолированных» и ничем не отвлекающих приложений. Немногие осознают, что эта функция существует в Linux уже так давно», говорит Кишор Бхаргава [Kishore Bhargava], ветеран индийского сообщества FOSS.

И пока некоторые пытаются освоить эту функцию, она претерпевает тщательнейший пересмотр на своем родном рабочем столе. Виртуальные рабочие столы более заметны в Ubuntu Unity, чем в Gnome 3. KDE, с другой стороны, подняла виртуальные рабочие столы на совершенно иной уровень.

«Я весьма впечатлен тем, как KDE расширил концепцию рабочего стола», отмечает Брюс Байфилд [Bruce Byfield], журналист, долгое время пишущий о Linux, добавляя: «Работая в KDE, вы теперь можете настроить рабочие столы для особых целей или местоположений, или изменить набор значков за считанные секунды. Эти инновации разработаны таким образом, что при желании вы можете и проигнорировать их, но если они вас заинтересуют, вы очень быстро их освоите. Я не могу назвать ни одного рабочего стола, ни под свободной лицензией, ни проприетарного, который сравнится бы с этой гибкостью KDE».

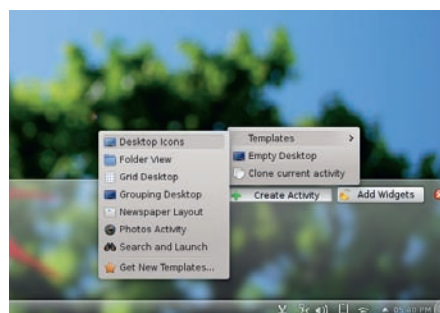


Шаг за шагом: Используем инновации Plasma Desktop



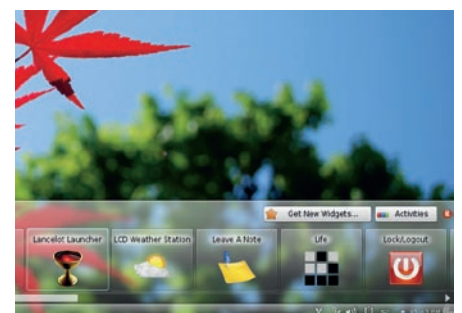
1 Добавляем виртуальные рабочие столы

Зайдите в Applications > Configure Desktop > Workspace Behaviour. Увеличьте число рабочих столов и щелкните по Different Widgets for Each Desktop.



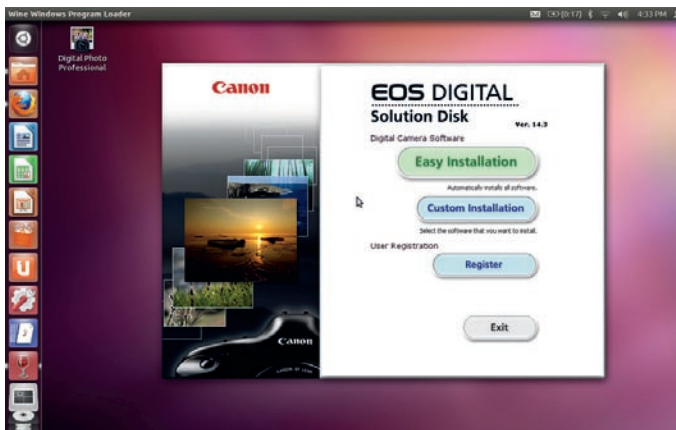
2 Создаем Activity

Перейдите в Activity Manager > Create Activity. Можете создавать Activity на основе шаблона, такого, как Desktop Icons. Щелкните по значку с гаечным ключом, чтобы редактировать имя и значок Activity.

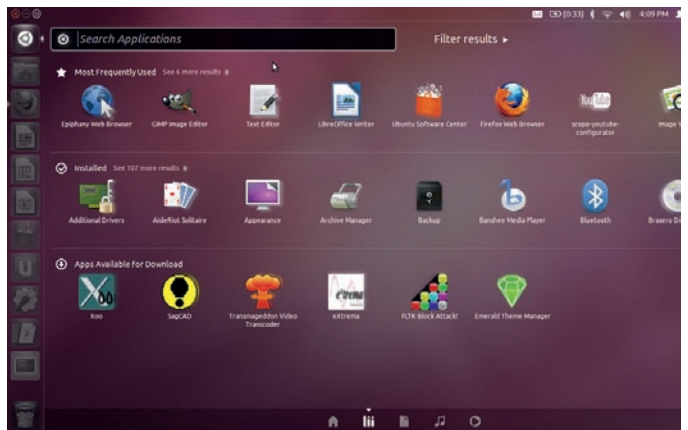


3 Настраиваем Activity

Вот оно! Пора настроить ее поведение. Можете добавлять виджеты, перейдя в Activity Manager > Add Widgets, а также сменить обои.



➤ Обратная совместимость *Wine* признана превосходящей более новые версии самого Windows.



➤ Одна из инновационных функций Unity Dash в том, что ее можно расширять, добавляя «линзы».

Украшения рабочего стола

Рабочие столы в виде куба, пульсирующие окна, тени и многое другое стали важной частью рабочего стола Linux уже более пяти лет назад. Архитектура сервера Xgl X, разработанная, чтобы использовать преимущества современных видеокарт, появилась в январе 2006. Она работала вместе с композиционным менеджером окон, например, с *Compiz*, для создания потрясающих эффектов рабочего стола.

И хотя *Compiz* появился уже после программы композиции *Quartz* из Mac OS X, он добавил украшений на все виды рабочих столов Linux, даже при заурядном оборудовании.

После появления KDE 4.2, собственный менеджер окон рабочего стола, *Kwin*, приобрел те же композиционные навыки, что и *Compiz*, а Gnome 3 переключился на менеджер композиций *Mutter*, который заставляет работать Gnome Shell. Но Ubuntu упорно цепляется за *Compiz* на своем рабочем столе Unity.

Управление пакетами

Один из самых распространенных мифов о рабочем столе Linux – это недостаток приложений. Благодаря намного большей экосистеме откры-

того кода, приложений в Linux не просто много – это одна из самых удобных для установки приложений операционных систем.

Главная инновация Linux – управление пакетами, и оно существовало задолго до появления магазинов приложений для смартфонов.

Благодаря таким приложениям, как *Synaptic*, *Zypper*, *YumEx* и другие, мы уже долгие годы находим и скачиваем приложения в Интернет.

Фактически, мы достигли в этом таких высот, что такие дистрибутивы, как Ubuntu, позволяют устанавливать приложения, не запуская менеджер программ. Тот факт, что вы притом можете компилировать приложения из исходника, свидетельствует о гибкости рабочего стола.

Байфилд добавляет: «В свободном ПО и ПО с открытым кодом меня нередко восхищало то, что, приступая к созданию приложения, разработчик зачастую старается сделать его как можно более полным». Он подтверждает свое наблюдение, указывая на такие приложения, как *Amarok*, *K3b* и *Caliber*, добавляя, что в этих приложениях имеются все функции, способные понадобиться пользователям для их задач.

«Например, в *digikam* есть абсолютно все необходимое для управления изображениями,

а *Caliber* является основным среди менеджеров электронных книг».

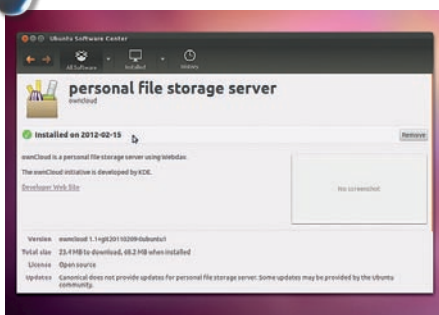
Поддержка приложений не-Linux

Еще одной важной инновацией, касающейся программ на рабочем столе Linux, является приложение *Wine*, позволяющее запускать приложения для Windows на рабочем столе Linux.

Проект *Wine* стартовал в 1993, задолго до того, как кросс-платформенные приложения стали нормой; а работа с виртуальной машиной тогда требовала больше умений и ресурсов, чем имелось у обычного пользователя рабочего стола. Изначально проект поддерживал 16-битные приложения Windows, но теперь он может поддерживать множество 32- и 64-битных приложений, таких, как сверхмощный *Adobe Photoshop CS3* и даже самые свежие игры, например, *Counter Strike: Source*. Даже с развитием простых в настройке альтернатив, таких, как *Win4Lin* и *VirtualBox*, *Wine* остается популярным выбором для пользователей рабочих столов, которым вынужденно требуются капризные приложения, работающие только в Windows, поскольку запуск приложений через *Wine* не требует лицензии Windows.

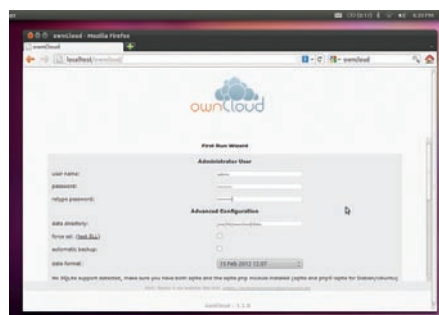


Шаг за шагом: Создаем собственное облако — ownCloud



1 Загрузите ownCloud

Пакеты для ownCloud имеются в репозиториях большинства дистрибутивов, и они настроят все нужные компоненты. Вы также можете скачать самый свежий релиз с сайта *ownCloud*.



2 Настройте его

Если вы установили *ownCloud* через менеджер пакетов своего дистрибутива, просто зайдите на <http://localhost/owncloud>, чтобы указать детали подключения, и можете приступать к работе.



3 Выведите его в режим онлайн

PageKite – прокси-сервер. С его помощью ваш *ownCloud* будет доступен отовсюду (загляните на <https://pagekite.net/wiki/Howto/GNULinux/OwnCloud/>).

Предприятие

Пусть кое-кто и сомневается в превосходстве Linux на рабочем столе, но в сфере применения на предприятии это бесспорный чемпион. Linux предоставляет инновационные решения для всех с ним работающих, обеспечивая максимальное время бесперебойной работы серверов. Понятно, что Linux соответствует гибкости предприятия благодаря тому простому факту, что здесь не делается искусственного различия между рабочим столом или сервером, домашним пользователем или бизнесом. Бхаргава, который использует Linux в качестве основной серверной платформы более 20 лет, говорит, что надежность и отказоустойчивость этих серверов вне конкуренции.

Он добавляет, что популярный стек приложений LAMP изменил «правила игры» для Интернета: «Web-приложения, которые плодятся, как мухи, обязаны своим появлением Linux. Среди них много социальных платформ и систем управления контентом. Благодаря Linux существуют приложения, среди которых ERP, управление отношений с клиентами, управление проектами и практически все, что только можно вообразить».

Регулярные обновления

В отличие от не слишком гибкого цикла поддержки, обычного для мира проприетарных программ, циклы поддержки дистрибутивов Linux подходят всем видам предприятий. И подобно управлению пакетами на настольных системах, применение заплат и установка обновлений – наверное, наименее заметная инновация Linux в сфере предприятия.

Также в отличие от мира проприетарного ПО, в Linux имеются и коммерческий, и бесплатный типы поддержки. На самом деле, в духе открытого кода, даже коммерческие дистрибутивы, скажем, RedHat Enterprise Linux, позволяют другим дистрибутивам использовать свои обновления уровня предприятия без оплаты услуг поддержки.

С течением времени программа обновлений в Windows улучшилась, но она и близко не подошла к надежности обновления программ в дистрибутивах Linux, где применение обновлений – дело пары щелчков мышью, да и этот процесс можно автоматизировать.

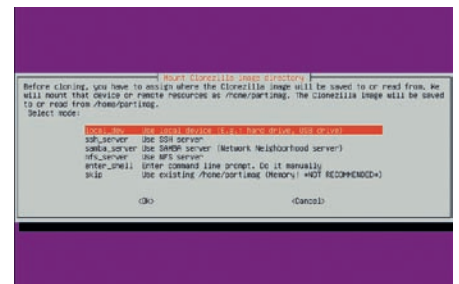
Надежная многопользовательская ОС

Linux начал свою жизнь как ОС для серверов и сетевых рабочих станций, что резко контрастирует с другими ОС наподобие Windows, которая возникла ради удовлетворения потребностей единственного пользователя, сидящего перед ней.

Обслуживание многочисленных пользователей, которые входят в систему локально или удаленно через SSH – врожденная функция Linux, а не приобретенная способность, как в Windows Server. Благодаря многопользовательской структуре Linux присуща большая степень безопасности, чем Windows. В Windows сидящий перед монитором пользователь имеет полный доступ к компьютеру. А вот в Linux для пользователей предусмотрена по умолчанию система разрешений на доступ к конкретным ресурсам компьютера. Парадигма пользовательских привилегий в Linux простирается от серверов до настольных ПК. Помимо этого, в Linux также имеется множество инновационных инструментов безопасности, которые дают ему преимущество перед Mac OS X. С помощью таких инструментов, как *iptables*, *fwbuilder*, *SELinux* и *AppArmor*, можно защитить дистрибутив самыми разными способами и на самых разных уровнях.

Интегрированная оболочка

Типичный пользователь рабочего стола не особо стремится что-либо делать из командной строки, но для большинства продвинутых пользователей это крайне важный и зачастую незаменимый ин-



Linux предлагает множество инструментов восстановления системы уровня предприятия для Mac и Windows.

струмент. И хотя в Mac OS X имеется весьма солидный набор инструментов командной строки, они далеки от той функциональности, которой обладают такие же инструменты в Linux. Вы можете запустить оболочку *Bash* на Mac, но в плане настройки и обслуживания в OS X найдется немало частей, которые нельзя редактировать вручную так, как это можно сделать в Linux. В Windows использование командной строки ограничено до такой степени, что порой вообще невозможно осознать, что она там есть.

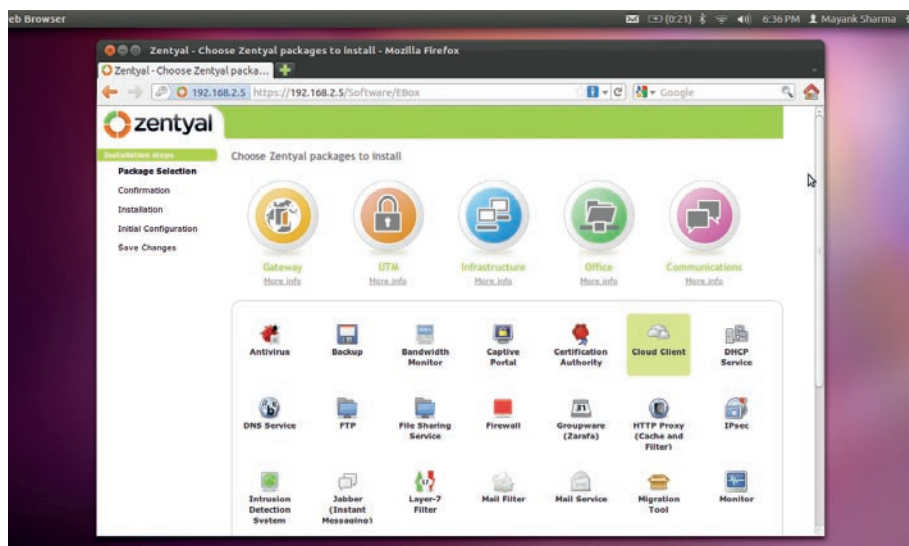
Даже базовые задачи командной строки, такие, как установка пакетов, остановка зависших приложений или запуск и остановка сервисов, невозможны в Windows. Но в 2006 Microsoft начал использовать *PowerShell*, где имеется набор команд и функций, аналогичных присутствующим в оболочке Linux.

Продвинутые файловые системы

Эффективная организация и хранение данных в файлах и обеспечение быстрого доступа к ним особо важны в сфере бесперебойной работы предприятия. И здесь Linux вновь превзошел своих конкурентов благодаря инновационным файловым системам – журнальной Ext3, а теперь и Btrfs (B-tree filesystem).

Появившаяся в ноябре 2001 года Ext3 предоставила пользователям преимущества журналирования операций, которые повышали надежность и устраняли необходимость проверки файловой системы после неправильного выключения. И, что самое главное, она позволяла обновить файловую систему ext2 и не требовала, чтобы пользователи делали резервные копии и восстанавливали свои данные в новой файловой системе. Современная файловая система Windows, NTFS, реализует способность к журналированию только через модули расширения.

Теперь, в Windows 8, Microsoft планирует заменить файловую систему NTFS новой файловой системой под названием ReFS (Resilient File System). У архитектур ReFS и Btrfs есть некоторое сходство – например, способность создавать мгновенные снимки копирования при записи [copy-on-write snapshots].



В дистрибутивах вроде Zentyal инструменты с открытым кодом промышленного стандарта для бизнеса настраиваются за считанные секунды.

Встраиваемые устройства

Linux и технологии открытого кода играют ключевую роль во встраиваемых устройствах с того момента, когда мир встраиваемых устройств перешел с микроконтроллеров на микропроцессоры, считает Сиджи Санни [Siji Sunny] – он уже несколько лет занимается интегрированием и разработкой встраиваемого Linux на различных устройствах.

Крис Симмондс [Chris Simmonds], основатель и ведущий программист 2net Limited, компании по разработке встраиваемого Linux, заходит дальше: «Linux изменил встраиваемые компьютерные технологии больше, чем любую другую область применения. Судя по числу задействованных экземпляров, можно утверждать, что Linux – основная встраиваемая ОС. Самая очевидная история успеха – Android, но не забывайте и о том, что Linux используется в большинстве домашних Wi-Fi-роутерах, в спутниковой навигации, в телевизионных приставках, в телевизорах последнего поколения, в принтерах и других устройствах».

По словам Симмондса, Linux начал внедрять инновации в мире встраиваемых устройств еще в 1997 с проекта Linux Router Project (LRP), ныне не развивающегося: тот был достаточно мал, чтобы уместиться на флоппи-диск, и превращал ПК в роутер для применения дома и в малом бизнесе. «Он был первопроходцем технологической создания компактных дистрибутивов Linux, в том числе Busybox, предлагающего командную строку и большинство обычных утилит Linux в аккуратном пакете объемом всего 100 КБ», поясняет

«Linux начал внедрять инновации со встраиваемого проекта Linux Router.»

Симмондс, добавляя, что Busybox имеется почти во всех современных встраиваемых устройствах Linux, кроме Android: там позаимствовали идею создания своего набора инструментов, toolbox.

LRP привел к появлению Linux на всех видах роутеров. По Симмондсу, дело тут в том, что «Linux



➤ Навигационное устройство TomTom, появившееся в 2004 году, было устройством ARM, применявшим Linux для графики, звука и ввода с сенсорного экрана.

был и остается очень хорошей платформой для сетевого роутера по причине его сетевой стековой памяти, ряда драйверов устройств и сетевых утилит». Еще одной инновационной функцией Linux, весьма способствовавшей его успеху в качестве встраиваемой ОС, является умение работать с большим количеством платформ. Резюмируя историю о портировании

Linux на большое число компьютеров разных типов, Симмондс говорит, что попытки сделать код Linux портируемым начались давно, с портов в рабочие станции Sun Sparc и Dec Alpha. «К 1996 году он работал также на Macintosh (PowerPC) и DECStation (MIPS RS4000), а к 1999 – на RiscPC (ARM). Трое последних, PPC, MIPS и ARM, образовали основу большинства современных встраиваемых устройств», объясняет он.

Решения нового поколения

Среди инновационных решений, предлагаемых разработчиками встраиваемого Linux по всему миру, Санни выделил проект Kivy. Это библиотека с открытым кодом, созданная на OpenGL ES 2 для разработки кросс-платформенных мультитач- и основанных на жестах приложений для устройств Linux, Mac, Windows и Android.

Другой инновационный проект, упомянутый Санни – ViSparsh, для помощи слабовидящим. Проект объединяет ряд устройств, например, компьютер PandaBoard и Kinect (устройство ввода от Microsoft, чувствительное к движению), со средой OpenNI и инструментом *mbed* для быстрого прототипирования с микроконтроллерами.

Роль Linux в сегменте встраиваемых систем растет с увеличением количества недорогих и открытых устройств. Одним из таких устройств, приобретающем все большую популярность у разработчиков, желающих поработать со встраиваемым Linux, является BeagleBoard xM. А недавно появился еще и компьютер Raspberry Pi, способный работать с несколькими дистрибутивами Linux.

Скрытые инновации

Помимо явных инноваций Linux на рабочем столе, есть и такие, что скрываются за кулисами – в ядре Linux.

«Одной из действительно замечательных инноваций, сразу же приходящих в голову, которая охватывает диапазон от встраиваемых устройств до настольных систем, является LLVM», говорит Майкл Ларабель [Michael Larabel], главный редактор Phoronix.com, который отслеживает все новые разработки в области ядра.

Ларабель добавляет, что инфраструктура компилятора LLVM лежит в сердцевине множества проектов ПО, таких, как драйверы OpenCL, графиче-

ские драйверы и компиляторы кода, и используется также множеством JIT-движков разных проектов типа Mono. Также он нашел интересные случаи применения в Android и даже в антивирусных сканерах.

Еще одна упоминаемая им инновация – *Galium3D*, которая, по словам Ларабеля, «позволила графическим драйверам Linux достичь зрелости куда быстрее, чем их двоичным конкурентам, способствуя большей степени открытости кода, оптимизации большей производительности и появлению новых функций, например, видео и компьютерной акселерации».



➤ Анонсируемый в последнее время планшет KDE Spark будет работать на Mer, одном из новых дистрибутивов Linux для мобильных устройств.

Чему следует научиться Linux?

Трудно выдвинуть довод против использования Linux на мощных серверах-мастодронтах для предприятия или на небольших встраиваемых устройствах. Но что касается настольных систем, то, невзирая на все инновационные функции, здесь остается простор для улучшений. Чего не хватает настольному Linux и что отделяет его от двух самых популярных в мире настольных операционных систем?

Восьмой основной релиз Apple, Mac OS X Lion, вышел в прошлом году. В нем более 250 новых функций, и компания называет его «своей лучшей ОС». Microsoft тоже трудится над завершением своего следующего релиза, Windows 8, с изобилием новых возможностей.

Так каким трюкам они могут нас научить?

Связь с OEM

Ну, тут все ясно. Вы не видите безумной давки в Dixons [сеть магазинов электроники в Великобритании, — прим. пер.] при каждом выходе новой версии Ubuntu или Fedora, потому что производители этих дистрибутивов не имеют прочных OEM-связей с основными производителями ноутбуков или настольных ПК.

Раньше, когда самые популярные дистрибутивы были результатом полуночного труда добротных или просто любопытствующих, OEM-связи были невозможны. Но даже сейчас, когда самые популярные дистрибутивы получают финансирование от мультинациональных корпораций, невозможно прийти в магазин электроники и купить компьютер с предустановленным Linux.

Вполне очевидно, что когда Linux появится в виде предустановленной системы на компьютерах, пользователи станут рассматривать его как полноценную опцию для рабочего стола, а не только как средство экономии денег. Заодно это поможет развеять расхожие предубеждения по поводу Linux — например, относительно медленной загрузки или недостаточной поддержки оборудования.

Улучшение комфорта пользователя

У Apple можно поучиться тесной интеграции настольных приложений с интерфейсом пользователя рабочего стола. Честно говоря, такое стало появляться только в последней версии Ubuntu, которая интегрирует различные сервисы Canonical с рабочим столом Unity.

Интеграция всегда будет более сложной проблемой в мире открытого кода Linux, чем, скажем,

«Unity, Gnome 3 и Plasma демонстрируют поиск инновационных решений.»

в жестко контролируемой среде Mac. Только на то, чтобы приложения Gnome или KDE притерлись к другим рабочим столам, ушло несколько лет. Мы не против применения к одной проблеме нескольких разных подходов, потому что это во многих случаях ведет к лучшей реализации. Проблема возникает, когда проекты начинают разрываться в разные стороны без особых видимых причин.

Улучшение приложений

Одной из наиболее часто упоминаемых причин большей популярности Mac по сравнению с другими платформами являются его дружелюбные к пользователю приложения. *Time Machine*, утилита резервного копирования Mac OS X — явный фаворит пользователей рабочего стола благодаря простоте использования и приятному интерфейсу.

В связке с другим инструментом, *Migration Assistant*, резервное копирование *Time Machine* пригодное для восстановления системы на Mac. Теоретически, то же возможно и на рабочем столе Linux, и должно дать вам лучшее управление; и, вероятно, может быть автоматизировано, если вы готовы приложить руки к CLI. Но реальной графической альтернативы для *Time Machine* нет.

Одна из дружелюбных к пользователю функций в Mac OS X Lion называется *AirDrop* — она позволяет отправлять файлы беспроводным способом на другой Mac, а также автоматически создает одноранговое зашифрованное соединение с тем Mac'ом, на который вы хотите отправить файлы.

И у Mac OS X Lion, и у Windows 8 есть новые инструменты обновления и восстановления системы без мороки со средствами установки. Помимо полной переустановки, *Lion Recovery* также может вос-

создать данные из резервных копий *Time Machine*. Подобным же образом Refresh and Reset, функция Windows 8, позволяет быстро и безопасно уда-

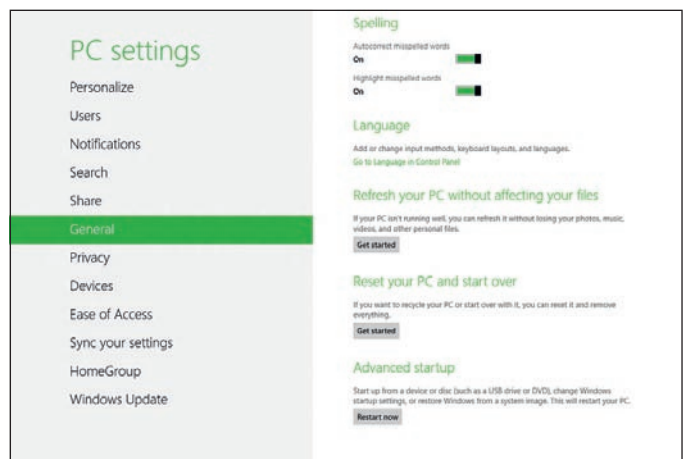
лить имеющуюся систему и установить новую, сохранив при этом свои данные и настройки.

Есть еще функции, которые эти операционные системы добавили в существующие приложения. Например, *Lion* добавляет к текстовому редактору *TextEdit* способность управления версиями. Он может также сохранять состояние приложений, когда вы из них выходите. Это особенно удобно, если у вас открыто несколько приложений и вам нужно перезагрузить компьютер. С помощью этой функции вы перезагрузите компьютер и возобновите работу со всеми приложениями в том виде, в котором вы их оставили. Как мы уже отметили, некоторые из этих функций воспроизводимы в Linux, но такое требует больше умений, чем имеется у среднего пользователя рабочего стола.

Поэтому нас и вдохновляют последние разработки на рабочем столе Linux: любите вы их или ненавидите, Ubuntu Unity, Gnome 3 и KDE Plasma демонстрируют поиск их разработчиками инновационных решений для улучшения комфорта пользователя. Пусть им потребуется время для достижения этой цели, но они ее достигнут, а по пути еще и научат проприетарных родственников парочке приемов. **LXF**



► *Time Machine* делает локальные снимки и автоматически связывает их с резервными копиями на внешнем носителе, при его подключении.



► Функция Refresh and Reset в Windows 8 идеальна для использования в корпоративной среде.



YOUNG
REWIRE
STATE

Хак в будущее

Мероприятие, где детей учат программировать и взламывать правительственные данные? **Джонатан Робертс** взглянул пристальнее.

За последние несколько месяцев в британском компьютерном образовании произошла целая серия волнующих событий, и не последнее место в ней занимал выпуск Raspberry Pi в продажу. Усилия по улучшению преподавания ПК уходят далеко в прошлое, и одним из наиболее интересных проектов является Young Rewired State.

Это захватывающий День хакера – Hackday (на самом деле неделя, не меньше!), который происходит раз в год: юные программисты со всей страны встречаются с принимающими фирмами

и учителями, чтобы познать тайны программирования, работу с открытыми данными и деятельность цифрового бизнеса.

«Юные программисты встречаются с фирмами и учителями.»

Усилия по организации Young Rewired State 2012 набирают обороты – особенно это касается поиска спонсоров, принимающих компаний и но-

вых «YRS'еров». Помня об этом, мы собирались поговорить о том, что туда входит и кто занимался Young Rewired State ранее, чтобы понять, что оно собой представляет.

Мы считаем, что невозможно устоять перед этим уникальным сочетанием открытых данных, духа сообщества, свободного кода и социальной значимости, и надеемся, что после прочтения статьи у вас сложится такое же впечатление.

А на самом деле, мы надеемся, что вы захотите поучаствовать, будь вы студентом, преподавателем или организующей стороной.

Чтобы вы поняли, что такое Young Rewired State (YRS, Rewired State для молодежи), вам придется узнать, что такое Rewired State [можно перевести как «перезагрузка всей страны», – прим. ред.], поскольку на сайте YRS охарактеризована как «рука благотворительности» у Rewired State.

Для этого мы поговорили с Эммой Малквини [Emma Mulqueeny], руководителем обоих проектов. Она рассказала, что «Rewired State – это (рас-тущее) сообщество из более 600 разработчиков старше 19 лет, и 200 разработчиков, возраст которых 18 лет или младше».

Такая сеть, по словам Эммы, выросла «из движения разработчиков, которые желают, чтобы правительство открыло свои данные». Данная мотивация означает, что многие из разработчиков в этой сети интересуются куда большим, нежели просто общение – они заинтересованы в «программировании страны к лучшему [coding a better country]», как говорит слоган проекта. С этой точки зрения, основной задачей Rewired State является организация Hackday и обеспечение обучающих программ для программистов.

Первый их Hackday, National Hack The Government Day, был проведен в 2009, и Эмма назвала его ошеломляюще успешным. «Во многом, – говорит она, – мы заслужили уважение тем, что подстегнули создание data.gov.uk». Поскреба по сусекам правительственных сайтов и попрошайничая у различных организаций, они создали базу данных, связанную с документами Кабинета министров, и в 2009 это был единственный открытый источник правительственных данных. То есть, по крайней мере, пока в том же году не запустили data.gov.uk.

Мероприятие, которое проводилось в офисах газеты *The Guardian*, объединило 80 разработчиков. Они работали с 10 утра до 6 вечера, создавая примерно 30 работающих проектов, которые затем были представлены СМИ и официальным лицам.

Один из созданных проектов был переписанным сайтом Active Places, с поддержкой для пользователей-инвалидов и мобильных устройств. Уже после восьми часов разработки всего лишь тремя разработчиками, большинство участников согласились с тем, что сайт существенно лучше оригинала, затраты на который составили 5 миллионов фунтов стерлингов.

После старта в 2009 Rewired State стал проводить больше Дней, а на этот год их запланировано еще больше. В 2011, наряду с организацией другого National Hack The Government Day [Общественный день хака правительства], они запустили Rewired State: Parliament, нацелившись на парламент. Данное мероприятие породило несколько проектов, подчеркнувших важность и открытости правительственных данных, и наличия программистов, имеющих достаточную заинтересованность и осознающих, как употребить ее с пользой. Возьмем, например, Who's Lobbying [Кто лоббирует]. Этот проект «впервые показывает, какие группы чаще всех выступают в парламентских комитетах». Или Little Ben, мобильное приложение, которое информирует пользователей, какие законы слушаются в парламенте, и дает им возможность проголосовать, причем парламентарии получают уведомления о ходе голосования – это позволит парламентариям обосновать свое решение.

Два этих проекта, которые способствуют большей прозрачности правительства и поощряют активность сограждан, показывают суть проекта Rewired State: программирование к лучшему.

Young Rewired State

Так как же появился Young Rewired State? Эмма рассказала, что как бы ни был успешен первый Hackday, «на мероприятии явно не хватало молодежи». И чтобы исправить это, было организовано специальное мероприятие, которое проводилось в лондонском офисе Google: Young Rewired State. Повестка дня была такой же: открытие правительственных данных и разные операции с ними, но только для молодежи.

На мероприятие, которое проходило в выходные, им удалось получить 50 подростков в возрасте от 15 до 18 лет. Разброс умений участников был очень велик; согласно комментарию в блоге после мероприятия, одни были искусными разработчиками, другие уже стали успешными предпринимателями, но у остальных были только базовые знания программирования и страстное желание узнать больше.

Нетрудно вообразить, что при столь широком спектре способностей проекты, созданные в YRS, будут стараться произвести впечатление на том же пути, что и проекты старших коллег. Отнюдь нет.



Эмма Малквини, исполнительный директор Rewired State и Young Rewired State.

Фото предоставил Пол Кларк (Paul Clarke)

Среди созданных ребятами проектов были, например, следующие:

- » инструмент сопоставления уровня тяжких преступлений и результатов экзаменов;
- » поисковая машина по университетским курсам;
- » база данных досуговой активности;
- » сервис, предоставляющий данные о расписании автобусов в реальном времени;
- » поиск работы с требуемым опытом.

По приложениям очевидно, что применение собираемых ими данных четко ориентировано на чаяния молодых. Сюрприза тут нет, но это показывает, как важно, чтобы программировать умели самые широкие слои населения.

Каждая демографическая группа имеет свой собственный набор проблем, о которых другие могут даже не задумываться. Если программирование останется делом умников, то технологии будущего не учтут интересы молодежи, женщин, старшего поколения, и многих других.

Аванпосты

Первый YRS был крупным успехом. С тех пор ежегодное мероприятие YRS продолжали проводить, но уже не под одной крышей, а распределенно, и длится оно уже не один уикэнд. По всей стране есть центры, где местный бизнес оплачивает помещение для детских групп, выпускников YRS и преподавателей Rewired State на целую неделю. Владельцы предприятий обеспечивают доступ к Интернету и преподавание, помогая детям определиться с проектом, разработать его и учиться по ходу работы. В конце недели все центры перемещаются в один пункт для дня презентаций.

Чтобы узнать, как это – принимать у себя центр YRS, мы поговорили с Гарри Хэрролдом [Harry Harrold] и Рупертом Реддингтоном [Rupert Redington]. Они владеют Neon Tribe, агентством web-разработок в Норидже и Норфолке, и сотрудничали с YRS с первого же года.

Мы начали с вопросов, как получилось, что они вообще попали в YRS. Гарри рассказал нам, что они взяли на работу школьника на неполный рабочий день, после того, как он набрался с ними опыта. Они послали его на YRS 1 при Google «исключительно потому, что он усердно потрудился; мы подумали, что он заслужил попытку, а YRS – это такая штука, за которой стоит гнаться».

Время — деньги

Главный вопрос, которым задаются компании, раздумывая, становиться им принимающей стороной или нет – не повредит ли это их производительности за неделю? В конце концов, для небольших компаний предоставление помещения и преподавания группе школьников на всю неделю – это серьезные затраты.

Именно в таком положении был Neon Tribe. Когда компания впервые провела неделю YRS, в ней было всего шесть сотрудников, и Гарри оценил, что им это обошлось примерно в три чело-

веко-недели – выходит, на мероприятие бросили половину персонала. Нам это показалось большой жертвой, но Гарри приуменьшил влияние мероприятия на производительность компании: «Другая половина [персонала] работала над долгосрочным проектом, и это был август, а в августе все равно сложно продвигать проекты, так что это был не конец света».

Итак, очевидно, участие вполне по силам и небольшой компании, без подрыва ее плановых оплачиваемых работ.

Зачем участвовать?

То был ответ на вопрос «как», а на вопрос «зачем» мы получили от Гарри и Руперта два разных ответа. Гарри сказал, что они «в общем симпатизируют мысли, что образование – штука хорошая».

«Руперт – бывший учитель, и наш первый проект в этой итерации бизнеса был разработан для образовательного телеканала. По сути, длительное время сферой нашей деятельности были общественные и благотворительные проекты, так что, полагаю, мы являемся социально ответственной организацией».

Участие приносит и чисто коммерческую выгоду. Гарри отметил, что они «хотели, чтобы в одном ряду с центрами цифрового творчества – Лондоном, Манчестером и Брайтоном – упоминались Норфолк и Норидж».

Руперт безусловно разделяет отношение Гарри к образованию, и согласился, что желание «повысить мастерство» молодежи вызвано важными коммерческими соображениями. Но для него это вторично.

А наиважнейшим он считает «показать молодежи, что прозрачность данных действительно важна для сознательного гражданина. Нужно помочь молодым преодолевать FUD [Fear, Uncertainty and Doubt – страх, нерешительность и сомнение]. Что мне нравится в Rewired State и его расширении Young Rewired State – оно собирает толковых людей и добавляет им здравого и обоснованного неуважения к власти и умения видеть чуть дальше всякой ерунды».

Такова была их мотивация, и когда к ним пришла Малквини и сказала: «Мы намерены сделать это общенациональным», Гарри понимал, что внутренне они уже созрели. Опыт первого года воодушевлял, а философия YRS пронизывала все аспекты их мировоззрения – принять решение было просто.

Типичная неделя

Согласие стать принимающей компанией означало предоставление своего офиса группе детей на целую неделю в августе. Поскольку они – компания разработчиков, и раньше уже участвовали в Rewired State, члены Neon Tribe всю неделю выступали также и в роли преподавателей.



➤ Майло Мордаунт, учащийся в Young Rewired State, фанат *Minecraft* и начинающий хакер.

В разных центрах неделя проходит по-разному, но, например, в Neon Tribe программированию отведено не все время. В разговоре с нами Гарри и Руперт подробно рассказали о неделе в 2011 году, охватив все: от планирования, сбора данных и программирования до финальной презентации.

В 2011 году первый день прошел в размышлениях о людях, ситуациях, в которых они находятся, и проблемах, с которыми они смогут столкнуться, выдвинув конкретную идею проекта. Именно преподаватель должен обеспечить, чтобы группа придумала идеи, пригодные к реализации в отведенный срок, с учетом доступных группе подростковых умений.

Если верить Руперту, ограничения однонедельного формата – важная часть YRS. «Мы постоянно имеем дело со сложными и хорошо отлаженными программами, и люди часто решают, что в своем первом программистском проекте они породят нечто вроде *World of Warcraft*. Им сложно представить себе задачу создания простой текстовой приключенческой игры».

«И вот преимущество короткого срока: все понимают, что такого они не успеют, но остаются реальными проектами, достойные трудов, которые следует завершить».

В 2001 году не было ясно, появятся ли подходящие идеи с ходу по итогам бесед, и преподаватели обратились к свежим газетам. Гарри рассказал: «Мы читали их, пока кто-то не сказал: "Норфолк



➤ Гарри Хэрролд из Neon Tribe участвовал в YRS с первого года.

выиграл 30 млн долларов на улучшение скорости широкополосного соединения», после чего все разом оживились».

«Дети жили в сельской местности, и одной из главных проблем у них была ужасная скорость соединения. Они сразу предоставили себе людей – себя – в ситуации и с проблемой: пытаться научиться программировать при кошмарной скорости Интернета». С точки зрения вовлечения детей и предоставления им конкретного проекта, в котором они могли что-то сделать, это было идеальной идеей.

Розыск данных

После определения темы, следующей задачей стал поиск каких-либо данных. Если знать, какие данные имеются, это поможет сформировать проект, прояснив, какие направления открыты по данному маршруту. Найденные данные исходили из сайтов проверки скорости Интернета и Ofcom. Это действительно любопытная часть работы, согласно Руперту. «Мы [как команда], полагали, что жизнь в сельском Норфолке чревата проблемами со скоростью Интернета, а они [дети] нашли официальные данные, где говорилось, что в Норфолке проблем со скоростью нет».

«Публичные данные Ofcom имеют склонность давать среднее по региону», а это очень широкий уровень, и данные никоим образом не совпадали с личным опытом детей. Тогда они решили отыскать альтернативные данные, которые подтвердили бы либо их опыт, либо правительственные данные. И они нашли определенную информацию на уровне почтовых округов, показавшую, что со скоростью Интернета есть проблемы.

Это случай «здроваго неуважения к власти» и грамотности, о которой говорил Руперт. Пройдя этот опыт, ребята уяснили, что официальные данные не всегда надо воспринимать как истину в последней инстанции, и если они не совпадают с вашим опытом, стоит копнуть глубже и проверить, что творится за кулисами.

Вдохновение Minecraft

На этом этапе у группы была задача и данные, и оставался вопрос, что же с ними делать. Первой мыслью было нанести данные на карту Google – популярную подборку данных; это мощный спо-

Открытые источники данных

Один из уникальных аспектов YRS (и Rewired State) – упор на использование открытых данных в качестве основы для хакерства. Но где же команды находят данные для хакерства?

Так вот, доступных данных существует огромное количество. В Великобритании правительство создало data.gov.uk, централизованный репозиторий всех публичных данных, и сделало его доступным, чтобы стать более прозрачным.

Сейчас сайт содержит более 5400 наборов данных, собранных по всем правительственным учреждениям, а также другим государственными органами и органами местной власти. Прекрасный пример данных, которые можно отыскать на этом сайте –

это сырые данные типа «выставление-затрат»: он содержит все транзакции свыше 500 фунтов для широкого спектра государственных органов, включая многие советы, Комитет по изменению климата и полицию Сэррея.

Еще один сайт, предоставляющий богатые данные – ScaperWiki. Как можно понять, целью ScaperWiki является сбор и извлечение информации, рассеянной по всему Интернету [scrape – англ. скребок]. Вообще источников много: сайт Public data от Google (<http://www.google.com/publicdata/directory>), тесты скорости соединения с Интернетом, университетские исследовательские работы... всего не перечислить!

соб отобразить географическое неравноправие. Технически задача не самая сложная, но предоставляет ряд серьезных проблем для школьников: им нужно изучить API, их применение и обращение с данными, чтобы они работали при ограничениях API. Однако, прежде чем погрузиться в труд, они отвлеклись на разговор о своих любимых занятиях за компьютерами. Один из учеников упомянул, что любит играть в *Minecraft*, крайне популярную игру (или даже «образ жизни», как повторяет наш Майк Сондерс), в которой вы населяете мир кирпичиками, и из их сочетаний строится практически все что угодно – включая целые страны, континенты и миры.

Так и возникла идея о представлении этих данных на карте *Minecraft*. Очевидно, что при большой популярности *Minecraft*, такой формат Великобритании с наложенными на нее скоростями Интернета привлечет гораздо больше внимания, чем просто очередная подборка с Google Map.

И это куда более дерзкий проект, как сказал нам Гарри: «Сначала нужно построить *Minecraft*-карту Великобритании. Но так вы не увидите данных – значит, нужно выработать способы программно размещать что-нибудь на карте *Minecraft* и переводить свои данные в данные *Minecraft*».

«Естественно, – говорит Гарри, – сделал это, вы сможете установить расширение, позволяющее полетать над картой Великобритании, просто в порядке дополнения».

Что же получилось? Карта Великобритании, которую можно облететь и на которой каждый почтовый округ представлен башней. Высота башни обозначает скорость широкополосного доступа к Интернету в этом районе, и если она превысит определенный порог, на вершине башни загорается свет. Это область с хорошей скоростью.

Следует признать, что на Гарри и других преподавателей эта идея произвела впечатление: «Получилось пересечение знаковых понятий: у вас есть *Minecraft*, увязывающий открытые данные – гениально, черт возьми».

Оставшаяся часть недели прошла за изучением принципов построения в *Minecraft* – кое-где ученики знали больше, чем преподаватели – и объединении кодов вместе, чтобы все заработало.

Финальные презентации

По окончании хакерства (чтоб хотя бы работало!), все, что осталось сделать на неделе YRS – отправиться на национальное собрание, где встречаются все центры, а учащиеся представляют свои работы. Согласно Гарри и Руперту, этот день презентаций не менее, если даже не более, важен, чем любой другой аспект YRS.

Гарри сказал нам: «Для молодых людей, которые по стереотипу являются технарями, для них оказаться в центре внимания – это момент, когда можно блеснуть в любимой области. С технарями такое случается нечасто – в спорте, музыке, театре это бывает, а в науке – нет».

«Важно даже просто показать детям, что их навыки имеют ценность, ведь все эти типы в костюмах пришли сюда, чтобы их послушать; важно показать им, что они не одиноки».

Точка зрения преподавателя

Последний элемент, который обеспечивает успех YRS – наличие хороших преподавателей. Они не обязательно приходят из принимающих компаний: часто там оказываются разработчики из более широкого сообщества, принимавшие участие во взрослых мероприятиях Rewired State.

В случае Нориджа в 2011 году, одним из преподавателей был Крис Хит [Chris Heath], местный web-разработчик. Он рассказал нам, что «ему становилось все интереснее Дни хака, но в его городе ничего такого не затевалось», и когда Neon Tribe организовала Rewired State, он «ухватился за этот шанс».

Одно повлекло за собой другое, и в конце концов его самого попросили помочь с YRS. Поскольку

Крис работает внештатно по договорам, он сказал, что «был счастлив на недельку отвлечься от работы ради такого мероприятия».

Опыта учителя у него не было, но он пояснил, что «это выглядело не как обучение, а скорее как работа в команде, но с большой долей руководства», и не вызвало больших проблем. По завершении финального дня презентаций (Крис уверяет, что «большая часть увиденных там презентаций были более высокого уровня, чем на настоящем производстве!») он теперь «фолловит многих преподавателей и детей в Twitter, и все остаются на связи. @Майло пишет мне в Твиттер, и я отвечаю, как только смогу. Я думаю, он знает, что всегда может спросить у меня совета».

Конечно, угодив в центр внимания, человек может растеряться. Если не будет правильной поддержки и одобрения, можно все испортить.

Однако Руперт поспешил подчеркнуть, что «презентация – это вполне благоприятная атмосфера, даже если она не задумывалась как таковая. Никто не прилагает специальных усилий для поддержки других, но все поддерживают друг друга просто потому, что делают общее дело. Есть чувство, что все мы вместе».

Презентация *Minecraft* 2011 года вышла настоящим победителем. Участники не только получили награду Best In Show за свой хакинг данных, но и сделали прекрасную презентацию. Она началась с разговоров о проблемах со скоростью Интернета: «Многие кивали, так как никто не верит в то, что у них скорость достаточно высока», вспоминает Гарри.

Когда они заговорили про карты, в аудитории опять закивали: «Все думали, что знают, что произойдет – “ах вот как, они разместили данные на карте, неплохо», однако, когда все открылось... они были сказали “все карты примерно одинаковы”, и вдруг облетели Великобританию над башнями, обозначающими скорость! Кто-то позади меня выдохнул: “Это же *Minecraft*!” Это был очень волнующий момент».

Точка зрения учащихся

Вот так закончился YRS 2011. Но мы не хотим завершать статью, не рассказав про точку зрения учащихся. В конце концов, YRS целиком посвящен именно им. Гарри познакомил нас с Майло Мордаунтом [Milo Mordaunt], одним из участников проекта *Minecraft*.

Он рассказал, что увлекся компьютерами, когда команда Neon Tribe (там работает его мама) познакомила его с Linux – увидев вращающийся куб *Compiz* на их компьютерах, он захотел такой же, и на его компьютер поставили Ubuntu и сделали нужную настройку.

Майло поясняет, что увлекся настройкой «этой штуки». «*Compiz* сделал хорошее дело – я подсел на настройку этой штуки, и сейчас я подписчик LXF и тестер Arch Linux, и все благодаря вращающемуся кубу!»

Когда мы спросили его конкретно про YRS, он ответил, что «это здорово – собирать вместе людей со сходным образом мыслей, ведь в этой нишевой области их (сравнительно) мало».

Продолжая рассуждать об YRS, он указал, что его умения «сейчас более полезны», и что у него «намного больше желания учиться, ведь он видел, что его ровесники пишут приложения и веб-страницы!»

Про финальный презентационный день Майло сказал: «Это было лучшим! Реально классные ребята просто стояли и говорили о проектах, которые они, сами знаете, состряпали всего за неделю. Причем ценные проекты!»

После YRS

После того, как все сделано и прибрано, учащиеся, преподаватели и предприниматели расходятся своими дорогами. Но это не означает конец дружбе и связям, созданным в течение недели. Большая часть сетевых связей поддерживается в Twitter – преподаватели предлагают свои советы, когда их бывшие ученики в них нуждаются, а многие учащиеся обсуждают идеи и продолжают поддерживать друг друга. Майло рассказал, что у него возникла идея «не так давно, насчет вроде как социальной сети для идей, и я получил большую поддержку в Твиттере от разных людей с YRS». **LXF**



➤ Руперт Редингтон считает, что у людей должно быть здоровое неуважение к властям.

Linux

КОСМИЧЕСКАЯ ОДИССЕЯ

Взгляд в Пространство

Дэвид Хейворд всегда мечтал стать космонавтом. В ожидании этого дня он пользуется Linux, приводя свой космос в порядок.

Linux — явно зверь хитроумный; он не только процветает на наших настольных системах, но также применяется на огромном количестве мобильных устройств и в серверных комнатах по всему миру. Однако вы, быть может, не в курсе, что Linux регулярно используется и в обстановке, весьма непохожей на дом, офис или машинный зал: Linux добрался и до сумрачной бездны, именуемой космосом.

Это кажется научной фантастикой — так и хочется процитировать легендарного капитана Керка [Captain Kirk]; однако за последнее десятилетие в авиации и космической отрасли произошёл переход от общепринятых ОС или специализированных встроенных систем к вариантам Linux. В чем причина такого сдвига? Причин ему несколько, и самая важная — финансы. Стоит ли тратить астрономические суммы на лицензирование продукта или на наем команды разработчиков решений по встроенным системам, когда Linux доступен за малую толику этой суммы, а сопровождение продуктов Linux выступает более дешевым решением.

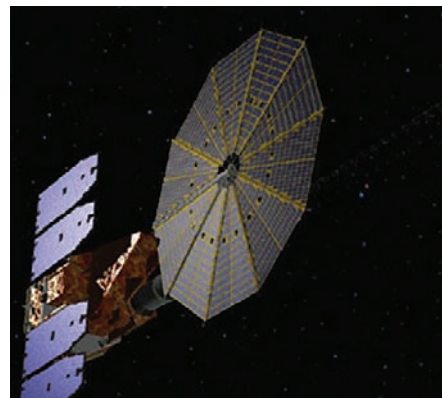
Другая причина, почему выбор пал на Linux, лежит просто в его хорошей работе. Очевидно, успешной космической операционной системе

требуются продвинутые алгоритмы, упреждающее планирование и более критический подход к выделению памяти, что известно как операционная система реального времени. ОС реального времени (RTOS) обязана удовлетворять потребности по времени у всех управляемых ею процессов, то есть она нацелена не на максимальную производительность, как наши стандартные настольные системы, но на должное согласование событий во времени. Естественно, ядро Linux — свободный ресурс, допускающий настройку — преобразовать в полнофункциональную RTOS легче, чем любую другую ОС. Справедливости ради отметим: чтобы сделать ядро настоящей RTOS, его придется несколько подрезать.

Итак, на следующих страницах мы бегло рассмотрим, какие космические программы используют Linux и какие будущие проекты включают нашу любимую ОС, держащую направляться туда, где не ступала нога человека.

ST8

Программа NASA нового тысячелетия, Space Technology 8 (ST8), похоже, столкнулась с проблемами из-за резкого сокращения финансирования, но отдельная технология этого проекта все еще су-



ST8 SpaceBus провозгласит новую эру в перевозке полезной нагрузки.

ществует и получает львиную долю бюджета. Это проект Dependable Multiprocessor (DM) — разработка аэрокосмического отделения Honeywell и Wind River (поставщик встраиваемых систем Linux). Идея состоит в создании компьютерной системы, которая будет более чем тысячекратно мощнее сегодняшних систем. С помощью готовых коммерческих систем (COTS) Honeywell и NASA хотят добиться системы, способной выстоять в суровых



Тестирование марсоходов различных конструкций, и все под управлением Linux.

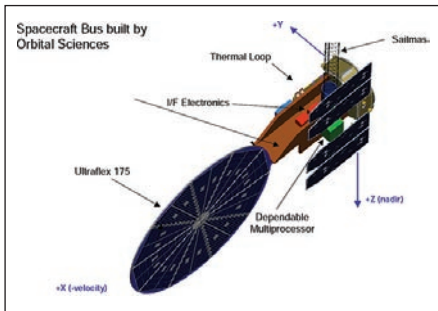
Следующий шаг

Будущее Linux в космосе выглядит особо привлекательным. Помимо всего прочего, нет ничего лучше хорошенького кризиса, чтобы компании перешли на свободные или более дешевые альтернативные операционные системы. С небольшим уклоном в сторону более дешевых, но более мощных компонентов – к примеру, смартфонов, производители ПО начинают подстраивать свою продукцию для

лучшего соответствия этим новым задачам. Одним из таких производителей является Red Hat, скорый релиз которого RH Enterprise MRG 2.1RHEMGR будет содержать RTOS-улучшенное ядро 3.0. Этот релиз привлек внимание Wind River, DoD и EKA как операционная система для будущих спутниковых кластеров и платформ для обработки данных. Нам, кстати, показалось забавным еще одно маленькое

открытие: мы обнаружили, что беговая дорожка на орбитальной станции фактически тоже управляется Linux на устройствах от NASA, с измененным ядром 2.6 от QinetiQ (bit.ly/yLVnzL).

По мере совершенствования ядра Linux растет и потенциал космических проектов, и, как следствие, в конечном итоге Linux приобретет знаменитость среди широкой публики.



➤ Это Star Ship Enterprise. Ой, нет, вообще-то это ST8 SpaceBus.

космических условиях, ежедневно перемалывая терабайты информации самым эффективным способом.

Космос, будучи не самым комфортной средой обитания, выдвигает массу требований к оборудованию. То, что прекрасно работает на Земле, может неминуемо давать сбои в космосе, в основном из-за излучения, от которого нас защищает атмосфера. Honeywell и Wind River используют слой ОС Linux (версия не оглашается) с интеллектуальными специализированными прошитыми микропрограммами, чтобы создать полуавтоматическую систему, которая анализирует самое себя, измеряя уровень радиации в форме заряженных частиц и проверяя свои подсистемы на предмет ошибок, вызванных этой радиацией.

Поверх этого стоят лабораторные функции обычной орбитальной или космической станции. Обычно их запускают в песочнице на Земле, затем передают в виде задачи на системы корабля; однако новое ПО системы DM, основанное на Linux, надо надеяться, сможет обрабатывать и анализировать собственные данные и принимать любые

решения о необходимости проведения осмотра без контакта с Землей и не дожидаясь ответа.

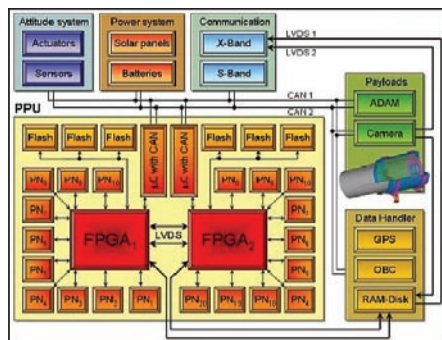
«Задачи космических полетов становятся все более сложными и требовательными, и важно строить бортовое компьютерное оборудование, способное поддерживать интенсивную обработку данных и выполнять должный анализ. По итогам демонстрации COTS в летном эксперименте ST8 ожидается, что применение технологии Dependable Multiprocessor в ближайшие годы приведет к беспрецедентным выгодам, в особенности в снижении времени разработки, цены и риска в будущих космических системах», рассказывает д-р Джон Сэмсон [Dr John Samson], главный инженер Honeywell.

Это довольно разумно, и хотя не дотягивает до полностью автономного уровня, какой был у HAL 9000, но, тем не менее, представляет собой большой шаг вперед в науке обработки данных, а также в компьютерных системах, умеющих самостоятельно принимать решения и действовать сообразно получаемой информации.

SpaceBus

Другой перспективный проект на базе Linux порожден описанным выше: им был ST8 SpaceBus. «Был» здесь значимо, поскольку, к сожалению, он был отложен, опять из-за досадных финансовых ограничений; однако дизайн и системы пригодятся в задачах будущего. SpaceBus – легковесный многоцелевой космический аппарат, основанный на платформе Orbital MicroStarT, которая применялась для космических научных полетов. Системой, которая «бы» управлялась и взаимодействовала с грузовыми модулями ST8, а также служила бы ОС для сетевого оборудования и для самых главных компонент управления кораблем, являлась платформа Linux Edition от Wind River.

К тому же, как показывает рисунок сверху, если корабль слегка повернуть, он станет вылитым Starship Enterprise.



➤ Схема кластера CPU PPU в X-Sat.

Беовульф в космосе?

Как, несомненно, многие из вас знают, под Beowulf мы понимаем термин, придуманный инженерами NASA Томасом Стерлингом [Thomas Sterling] и Дональдом Беккером [Donald Becker] для компьютерных кластеров, а вовсе не бородатого скандинавского безумца, который направо и налево крошил чудовищ.

Фактически, Beowulf – это кластер одинаковых соединенных сетью компьютеров, которые ис-

пользуют разные приложения, при этом позволяя распределять обработку по ресурсам всех компьютеров, задействовав для управления только основной ПК. Задача установки кластера довольно проста – можете даже попытаться осуществить это дома на паре свободных ПК [см. LXF148].

Такому типу вычислений есть множество применений, одно из которых – решить задачу о ширине полосы пропускания на микроспутнике X-Sat. X-Sat – это совместное предприятие Центра исследований спутниковых технологий (CREST), Наньянского Технологического университета (Nanyang TU) и Национальных лабораторий Сингапура. Его изначальной задачей было наблюдение и фотографирование Земли – слежение за уровнем моря, лесными пожарами, ростом населения и другими аспектами, касающимися окружающей среды; для этого он несет камеру с высокой разрешающей способностью, которая создает чрезвычайно подробные снимки и передает их на станцию обработки в Сингапуре. На борту есть два компьютерных устройства – ОВС (бортовой компьютер), обеспечивающий телеметрию, полезную нагрузку, связь и другие задачи спутника, и управляемый RTOS Linux VxWorks; и PPU (Parallel Processing Unit), который является кластером Beowulf из 20 206-МГц процессоров StrongARM, где стоит видоизмененное ядро Linux версии 2.4.4. Этот кластер был собран, чтобы помочь обработать более 80 Гб данных, которые X-Sat вырабатывает и передает на Землю за один день. Особо важен тот факт, что спут-



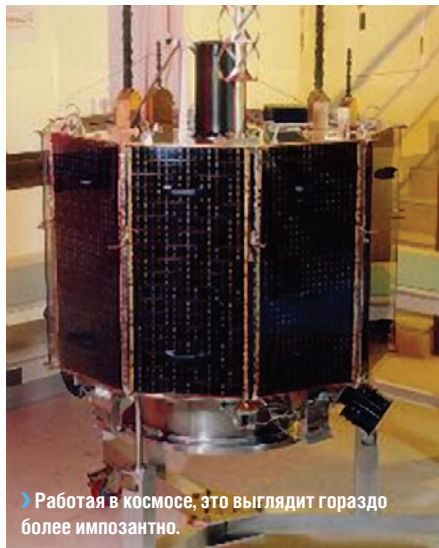
➤ Полезная нагрузка в X-Sat во всей своей красе – вот эта коробка слева.

ник имеет связь с Сингапуром только в течение нескольких минут за виток, и за короткое время нужно передать множество данных. Главная задача кластера – обработать собранную информацию и отослать только те изображения, которые представляют интерес для миссии, пока спутник пролетает над Сингапуром на высоте примерно 500 миль.

Проект NASA Omni Project

В плане коммуникаций, как ни странно, космические программы сильно отстают от положения дел на Земле. На наших скромных настольных системах мы можем заходить в Интернет, посылать друзьям электронную почту, проводить видеочат и играть в новые игры по высокоскоростному соединению. Однако для сотрудников международных космических станций (МКС) это непозволительная роскошь. Коммуникации между космическими челноками или с МКС в основном ведутся в закрытом режиме точка–точка по общепринятым и довольно старомодным технологиям; ну, а если на корабле будет доступен Интернет?

«Представьте, что на космолетах есть Сеть. Они могут общаться друг с другом, обмениваться данными и предупреждать других о том, что обнаружили своими датчиками», говорит руководитель проекта Джим Раш [Jim Rash]. Проект NASA Omni Project, разрабатываемый NASA в Центре космических полетов Годдарда, с 1998 года работает над размещением узлов IP на различных спутниках и МКС, используя средние процессорные мощности. Например, злополучный шаттл «Колумбия» был оснащен 233-МГц CPU ARM, 128-МБ ОЗУ и 144МБ SSD, с установленным Red Hat Linux 6.1, который поддерживал контакт с Землей с помощью IP, что превращало шаттл в интернет-узел (зеркало настроек NASA есть на МКС). А вот инициатива NASA Linux IP стартовала со спутника UoSAT-12. Этот микроспутник был разработан



Работая в космосе, это выглядит гораздо более импозантно.

и создан Surrey Satellite Technology Ltd, аэрокосмической компанией-подразделением Университета Сэррея.

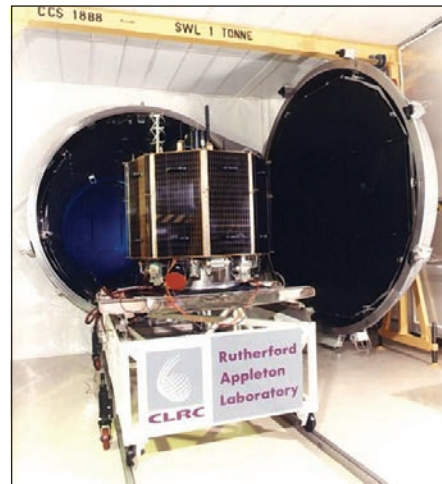
Запущенный в 1999, UoSAT-12 имел все необходимые компоненты, оборудование COTS и ядро Linux с IP-стеком, чтобы можно было загружать команды по IP, и в мае 2000 г. стал первым орбитальным IP-адресом, с экспериментальной 1-Мбит/с нисходящей связью с Землей. Вскоре UoSAT-12 пролинговался, смог синхронизировать свое время с NTP-сервером (tick.usno.navy

«Ваш код покинет Солнечную систему или приземлится на иной планете.»

mil) и проводить успешную FTP-передачу. В январе 2001 г. UoSAT-12 стал первым орбитальным web-сервером. К сожалению, сейчас у него нет IP-адреса и на нем не размещен web-сервер, однако он все еще летает, и если вы зададите вашему браузеру адрес www.n2yo.com/?s=25693, он покажет карту Google в реальном времени текущего положения этого пионера Linux в космосе.

NASA Open Source Portal

Орбитальные системы и операции, конечно, вне нашей сферы деятельности как обычных людей, но и мы можем получить немножко NASA на свои системы. Зайдите на NASA Open Source Portal, open.nasa.gov, сайт, где собраны главные элементы Open Government Plan [Открытый правительственный план] – прозрачность, участие и сотрудничество, и, как утверждает на сайте Open NASA Вильям Эшаф [William Eshagh], «Мы верим, что в будущем космические и научные системы будут строиться открыто и что code.nasa.gov сыграет в этом большую роль. Быть может, ваш код когда-нибудь покинет Солнечную систему или приземлится на другой планете? Мы работаем, чтобы воплотить это в реальность, и с вашей помощью достигнем этого». Хотя NASA и правительственная организация, она запустила в январе 2012 года от-



UoSAT-12 готовится к прохождению климатических испытаний перед запуском.

крытый портал, нацеленный на «повышение прозрачности, участия и сотрудничества Федерального правительства с американским народом». Эшаф рассказал *Linux Format*: «Мы приветствуем применение этих принципов в правительствах по всему миру, и недавно присоединились ко многим другим странам в Open Government Partnership. Великобритания, кстати, тоже в нем участвует (bit.ly/x5ez6d)».

Портал Open NASA успешно применяет таланты общества с максимальной пользой для самого же

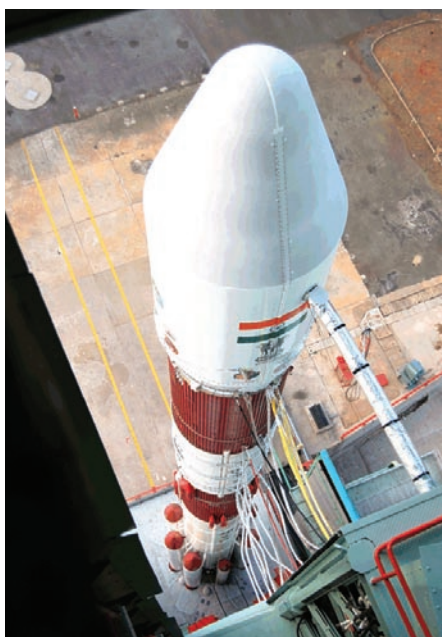
общества, так как NASA занимается не только отысканием инопланетян, но и проблемами, которые имеют непосредственное отношение к нам – на-

пример, изменение климата, распространение населения и мониторинг загрязнений. А уж если вы внесете вклад в любой из этих проектов, такое будет нельзя не упомянуть в вашем резюме!

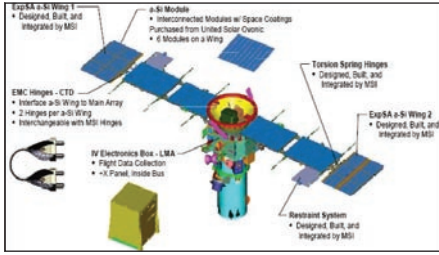
Проекты разнятся от создания трехмерной модели космического аппарата до утилит GIS-анализа для картографии лунной поверхности. Все они хорошо документированы, и их можно скачать либо с сайтов под управлением NASA, либо с Sourceforge, либо с GitHub, на множестве языков программирования и для различных платформ Linux/Unix. Все, что нужно – это отыскать проект, от которого у вас зачесутся руки, последовать по ссылкам, распаковать его и погрузиться в недра кода. Вот несколько примеров:

GMAT

GMAT, известный как General Mission Analysis Tool [Инструмент анализа общего назначения] – это открытая платформа, система независимой оптимизации траектории и дизайна. Другими словами, он помогает аппарату удерживаться на заданной траектории при движении по низкой орбите, орбите Луны и даже в дальнем космосе. GMAT использует язык скриптов, синтаксис которого похож на язык системы MathWorks MatLab, когда пользователь может анализировать и моделировать



X-Sat готовится к старту.

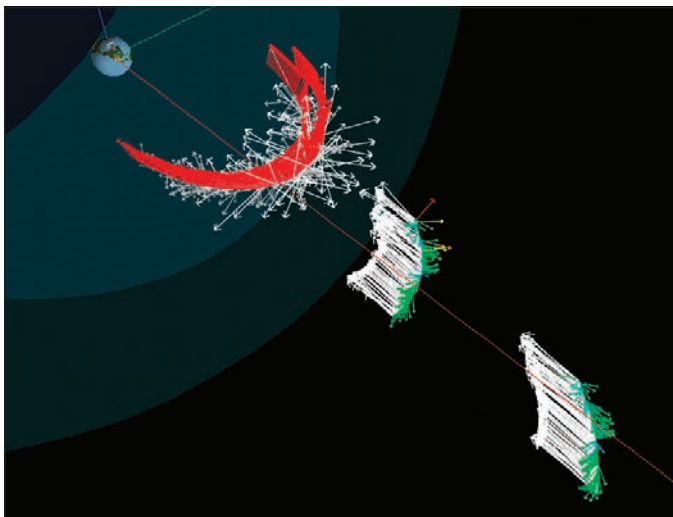


» TacSat-2, первоначальный дизайн CAD.

космические полеты, а затем применять этот анализ при моделировании, чтобы узнать, какие переменные требуют изменения, корректировки и т. д. Цель проекта GMAT – помочь разработать новую, более эффективную систему расчета космической траектории для космических двигателей и аппаратов следующего поколения. Если вы мечтаете послать космический корабль к последним рубежам или поэкспериментировать с траекторией МКС, зайдите на bit.ly/ACxklB за 64-разрядной сборкой для Linux; если вы любитель трудностей, вы всегда можете скачать исходный код и собрать GMAT самостоятельно, если посетите bit.ly/A14kt0.

ViSBARD

ViSBARD (Visual System for Browsing, Analysis and Retrieval of Data) – еще одно открытое, платформенно-независимое, интерактивное графическое приложение на базе Java; оно предоставляет способ одновременной визуализации многих скалярных и векторных величин, поступающих с приборов космического корабля. Количество информации тут практически непостижимо, а вот ViSBARD способен отобразить эти данные в виде понятной трехмерной цветовой модели. Ну, то есть, понятной, если вы – старина Альберт Э. Для остальных это просто очень красиво. Чтобы запустить программу, нужно только зайти на 1.usa.gov/ygW19x и нажать кнопку JAVA launch. Загрузка потребует времени, однако в Linux работает очень эффектно.



» Ну, что вы скажете — ведь совсем даже неплохо смотрится будущее устройство TomTom?

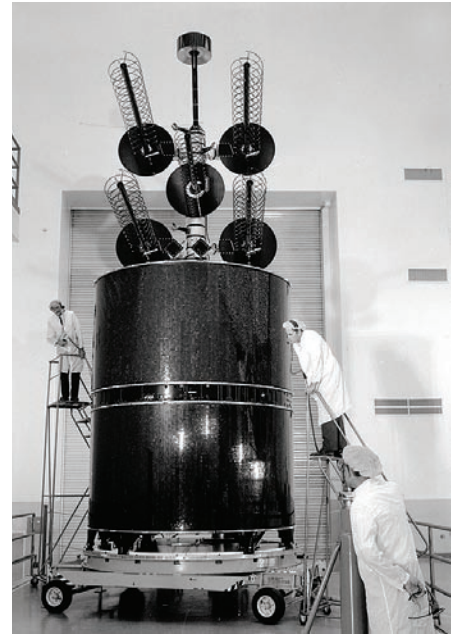
Это еще не все. Указав своему браузеру bit.ly/wzCXHo, вы попадете на International Space Apps Challenge, открытую глобальную разработку, где найдется место для каждого, кто любит «создавать, строить и изобретать новые решения» для глобальных задач. Эшэф рассказал *Linux Format*: «Благодаря этому событию, мы будем работать с космическими агентствами по всему миру, ускоряя разработку решений, связанных с улучшением правительства и главнейшими проблемами нашей планеты, включая влияние погоды на мировую экономику и истощение океанских ресурсов, но не ограничиваясь этим». Так чего же вы ждете? Вперед, пусть **LXF** гордится вами.

TacSat-1

В Америке NASA, похоже, является средоточием всего связанного с космосом, но это лишь маленькая капля в космической индустрии, поскольку уже появилось множество передовых инженерных компаний и организаций, которые занимаются космическими системами и компьютерным оборудованием, работающим под Linux. Самыми главными из них являются Министерство обороны США (Department of Defence, DoD) и Лаборатория морских исследований (Naval Research Laboratory, NRL).

У Министерства обороны сложная история милитаризации космоса и посылаемых туда объектов. Однако благодаря Договору о космосе времена ядерных орбитальных платформ якобы прошли. Сейчас DoD больше заботит обеспечение войск данными в реальном времени и помощь в командовании флотскими и военно-воздушными ресурсами. Из множества проводимых им экспериментов пользователям Linux интереснее всего TacSat-1, тактический спутниковый эксперимент – объединенный проект DoD и NRL, ранее известный как Operationally Responsive Space Experiment, занимающийся технологиями получения изображений.

Эксперимент состоял в разработке и создании небольшого спутника, менее чем за год, и предоставлении платформы для экспериментов, со-

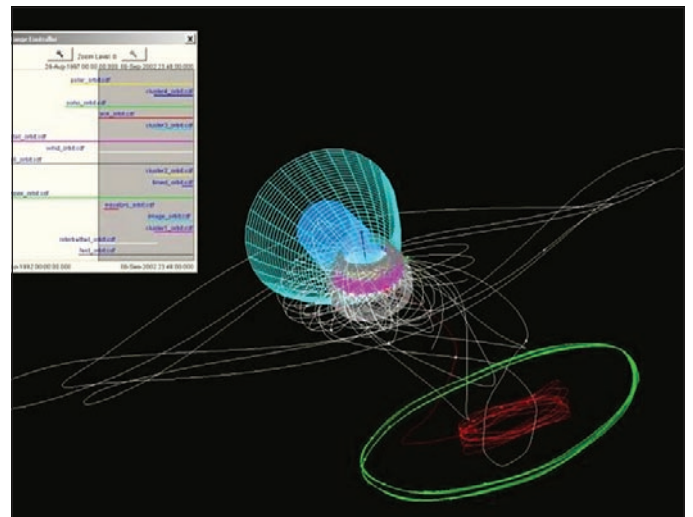


» TacSat-1 во всей красе. Какие шляпки...

четающих специализированные системы и готовые компоненты. Сообщение между компонентами строилось на базе ядра 2.4 Linux, при помощи коллекции скриптов *bash*, управляющих полезной нагрузкой между очень высокоскоростными интерфейсами. Эксперимент TacSat-1 выглядит примитивным для управления спутником, особенно когда мы уже поняли, что единственный способ управлять таким оборудованием – это RTOS, но он был нужен чисто для приобретения опыта и проложил дорогу куда более успешному TacSat-2.

TacSat-2

TacSat-2, после переименования JWS-D1 (Joint Warfighting Space Demonstrator) – это вторая фаза проекта NRL, который построен на технологиях, изученных в ходе TacSat-1. Спутник весом 415 кг был запущен с целью экспериментов по съемке,



» А вот такое происходит, если дать автору *Linux Format* порулить космическим аппаратом.

связи и GPS, предназначенных для облегчения мобилизации наземных войск. Эксперимент Target Indicator [Индикатор цели] работал на паре процессоров PowerPC на PCI-картах Motorola MPC8265s с 128 МБ ОЗУ, причем на свободном ПО, например Embedded Linux Development Kit (ELDK), созданным DENX Software Engineering, и ядре Linux DENX 2.4.25, которое можно скачать с bit.ly/A6Rvak.

ExoMaDeR

NASA – не единственный фрукт в космической вазе; Европейское космическое агентство (ЕКА) использует Linux по-своему. Возьмите, например, RTAI (Real Time Application Interface) для Linux – этот интерфейс позволяет писать приложения с четким таймированием. ЕКА разрабатывает RTAI для применения в прототипе планетохода ExoMaDeR, следующем поколении роботов для Марса.

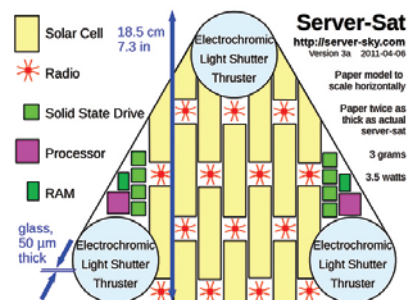
Используя RTAI на Linux, разработчики смогут программировать управление двигателями планетохода в реальном времени, чтобы он мог автономно изъездить неровный марсианский ландшафт вдоль и поперек, не управляясь с Земли – это важно, поскольку среднее расстояние от Земли до Марса около 250 000 000 км. Работу планетохода обеспечивает отдельная система XLuna, разработанная Critical Software (criticalsoftware.com). XLuna (eXtending free/open source real-time execUtive for on-board space Applications) – это ядро Linux, разработанное для широкого диапазона разнообразных космических

проектов, поддерживающее полезную нагрузку и авиационную электронику. XLuna работает как урезанное ядро поверх RTOS, которая называется RTEMS (Real-Time Executive for Multiprocessor Systems) и подразделена на две подсистемы, выполняющие задачи реального времени в тандеме: «Преимущество двух изолированных систем – возможность запуска на одной системе приложений с разными уровнями критичности. Тогда некритические компоненты разрабатываются быстро или просто портируются из предыдущей реализации, что приводит к сокращению бюджета и времени выхода на рынок».

XLuna поддерживает слой приложений, управляющий полезной нагрузкой на борту планетохода ExoMaDeR, и включает зонд для анализа, отслеживание траектории и внешние камеры для исследования окружающей среды, плюс плату CPU, твердотельный накопитель и модуль Wi-Fi для удаленной связи. Эта система также используется в беспилотных летательных аппаратах военного назначения и быстро интегрируется во многие из систем полезной нагрузки ЕКА для будущих полетов. Проект марсохода ЕКА стал несколько крупнее (см. рисунок), и его запуск запланирован на 2018 г. За подробностями обратитесь на bit.ly/xAM6c0.

Server Sky

Это интересный проект, который однажды сможет стать реальностью. Server Sky включает перевод на орбиту ультратонких стеклянных пленок, пре-



Теоретический вычислительный центр на орбите.

образовывающих солнечный свет в вычисления и связь. Тинсаты [thinsats], как их называют, будут питаться от большой солнечной батареи, и будут в постоянной связи с управлением посредством микроволн.

В космосе они охлаждаются почти до абсолютного нуля. Теоретически тинсаты смогут вести себя как большой орбитальный кластер Beowulf, предоставляя массам огромные объемы параллельных вычислений без большого потребления энергии и без проблем с окружающей средой, характерных для земных вычислительных центров.

Проект пока большей частью на бумаге, однако его автор Кейт Лофстром [Keith Lofstrom] готовит деловое предложение и ищет инвесторов. Хотите сыграть? Если да, зайдите на bit.ly/wPJMuz и ознакомьтесь с наукой, стоящей за орбитальным кластером. **LXF**



ExoMaDeR в действии. Навигации по неприветливой территории способствует Linux.



Новый дизайн ExoMaDeR, готовый к полету на Марс в 2018 году.

Поставщики космического Linux

Подумываете применить свои навыки Linux в космической индустрии? Вот несколько ссылок на сайты, которые мы обозначили как поставщиков Linux в авиации и космической индустрии. Все, что вам нужно – это послать им свое резюме, быть может, со ссылкой на открытое ПО для NASA, и кто знает, вдруг вы первым разработаете Linux на Марсе.

QinetiQ – bit.ly/yLVnzL: Эксперты по оборудованию

и приложениям, используемым в разнообразных средах, и одна из них – космос.

ETH Zurich – bit.ly/xK2lPE: Эксперты в области программ управления полезной нагрузкой и обслуживания для систем глобального наблюдения.

LynuxWorks – bit.ly/yTWoQU: Главный поставщик систем RTOS Linux, применяемых в слежении за Землей и других системах высокой доступности.

Wind River – bit.ly/yuA82p: Папаша систем Linux RTOS для космической индустрии. Подружитесь с этими парнями, и ваше будущее сделано из латинума в золоте [валюта во вселенной Star Trek, – прим. пер.].

Terma – bit.ly/z8tjyD: Основан в Дании, поставщик специальных решений для управления полетами космического проектирования.



MongoDB и Drupal

Drupal 7 Database API

Оптимизация фронтэнда

Использование модуля Skint

Drupal 7

Препроцессоры CSS, JS

Drush

Увеличение производительности Drupal

Development workflow

SSH

Темизация

Использование Git

AJAX и JS

Sphinx + Solr

Drupal Commerce

**MOSCOW
DrupalConf**

HTML5 и Drupal

Render API в Drupal 7

**2 ИЮНЯ
2012**

Debug in browsers

"Облачный" хостинг для Drupal

Development tools (IDE, CVS (svn, git))

Кто?

Разработчики, SEO-специалисты, владельцы веб-студий, системные администраторы, руководители интернет-проектов

Что?

Программирование, решение и настройка, использование модулей, темизация, дизайн и юзабилити

Зачем?

Будь на острие событий! Обменяйся идеями и опытом, повысь свой профессиональный уровень, проведи время в кругу единомышленников и друзей!

Как?

Стань участником конференции прямо сейчас:

www.drupalconf.ru





DragonFlyBSD

Предпосылки и история

Новые операционные системы возникают на наших глазах не каждый день. И, главное, не каждая из возникших систем активно развивается, а не забрасывается по прошествии недолгого времени или входит в состояние стагнации.

И в этой статье я расскажу о счастливой судьбе такой системы, пусть не завоевавшей мир, но вполне по жизни благополучной.

Как догадались многие читатели, речь пойдет о DragonFlyBSD, ответвлении FreeBSD, для которой отсчет времени пошел в июне 2003 года и за которой я следил с самого начала. И которую начал использовать с того момента, как она к тому стала пригодна.

«Железная» ретроспектива

Начало нулевых годов стало одним из переломных моментов в развитии информационной сферы. Вычислительные мощности обычных настольных персоналок достигли такой страшной силы, что сполна, а то и с лихвой перекрывали потребности подавляющего большинства пользователей. По крайней мере тех, чьим основным занятием было создание контента – до начала эры тотального его потребления оставалось еще несколько лет.

И тут обнаружилось, что, впервые с момента появления персональных компьютеров, софтверный сектор индустрии не смог выполнить свое сакральное предназначение – вытрясать пользовательские кошельки на перманентный апгрейд своих машин, что обеспечивало бы финансовые источники для дальнейшего развития сектора аппаратного. И производство базового компьютерного «железа» – процессоров и чипсетов, определяющих лицо платформы – если и не прекратилось, то резко затормозилось.

Конечно, время от времени производители продолжали бодро рапортовать об очередном повышении тактовых частот процессоров, увеличении их кэша до совершенно невообразимых объемов, некогда вполне достаточных для основной памяти, включении добавочных наборов команд под замысловатыми названиями, встраивании в чипсеты поддержки всего, чего можно, и даже того, чего, как недавно казалось, нельзя – например, 3D-графики. Однако накал страстей вокруг этого был не тот, что во второй половине 90-х. А отчеты о тестировании процессоров и материнских плат стали напоминать просмотр фотофиниша на стометровке.

Главное же, что, подобно рекордам в стометровке, достижения «камностроителей» все меньше волновали широкие массы пользователей. Ведь от медведя не убежит и олимпийский чемпион, а успеть за пивом в магазин перед его закрытием способен любой мало-мальски тренированный человек. Так и с компьютерами: настолько-пользовательские задачи в большинстве случаев стали

решаться средствами любого подручного «железа», купленного за пределами лавки древностей. А задачи «тяжелые» по прежнему требовали более чем всех ресурсов персоналок, и потому решались обычно не на них (по крайней мере, разумными людьми).

Правда, стимуляции пользовательского интереса для, производители в первые нулевые предложили два архитектурных решения, продаваемые как революционные. Первым (по времени) была архитектура Pentium 4. Она обеспечивала рост тактовой частоты процессоров, поражающий воображение пользователя, тянущегося к бумажнику. И к тому же перспективы роста казались тогда безграничными. Правда, насчет безграничности жизнь довольно скоро внесла свои коррективы. И разговоры о том, что эта технология позволит играючи достичь тактовой частоты в 10 ГГц, как-то стихли сами собой. Однако чисто случайно вышло, что гигагерцев, в отличие от прыжков, как раз и хватило на всех (кроме тех, которым, как уже говорилось, их не будет хватать всегда).

Вторая, столь же революционная, новинка – 64-разрядные вычисления. Вспомним третью статью цикла: каким прорывом в светлое будущее были 32-битные процессоры для PC! – те самые первые «трешки», которые сделали возможным портирование на эту архитектуру UNIX и, в конечном счете, появление Linux. Повторилась ли история на новом витке диалектической спирали?

Увы, отрицательный ответ был получен практически мгновенно. Потому что в те далекие уже годы аппаратура PC едва успевала за софтом – 32-битные ОС разменивали уже второй десяток лет своего существования, и приложений, использующих 32 разряда на полную катушку, было вдоволь. В описываемый же момент их в пользовательском сегменте просто не было по одной простой причине: невостребованности. К слову сказать, почти нет их и по сей день. Ибо единственная ниша пользовательских приложений, где 64 бита хоть как-то задействованы – параноидальная криптография.

Так что усилия «камностроителей» пропали бы втуне. Если бы не еще одно новшество, о котором я сознательно не упоминал ранее – Hyper Threading, то есть виртуальная мультипроцессорность. Каковая в некоторых (правда, весьма редких) задачах давала вполне даже реальный прирост производительности. Правда, он мало значил для работающих преимущественно интерактивно. Но весьма способствовал производительности труда тех, кто в силу врожденной лени отдавал предпочтение

всякого рода скриптам, пакетным заданиям и прочим средствам автоматизации.

Однако Hyper Threading был не более чем суррогатом истинной мультипроцессорности – своего рода мультипроцессорность для бедных, но гордых. И потому, сказавши А, производители процессоров неизбежно должны открыть рот для произнесения Б. То есть переходить к собственно мультипроцессорным конфигурациям в пользовательском сегменте.

Разговоры о двухпроцессорных пользовательских десктопах возникали неоднократно. Кое-кому из читателей памятно, как дешевенькие Celeron'ы первого разлива можно было вставлять в относительно не очень дорогие двухпроцессорные материнские платы, получая таким образом нечто вроде «народного суперкомпьютера».

Правда, первая волна «народной мультипроцессорности» была очень быстро пресечена производителем. Однако идея мультипроцессорности для народа продолжала витать в воздухе – никаким иным способом

создать впечатление прогресса уже не удавалось (к слову сказать, не удается и по сей день). И первый шаг в этом направлении сделала, насколько мне известно, IBM со свои-

ми процессором Power4 – в то время абсолютным рекордсменом по «чистому» (то есть тестовому) быстродействию. В том числе и благодаря тому, что он имел варианты с двумя и более процессорными ядрами в едином корпусе.

Сами по себе процессоры Power4 (как и пришедшие им на смену Power5) ориентировались на индустриальный сектор. Однако на базе их были созданы процессоры G5 – сердце тогдашних Mac'ов, имевших, в том числе, и двухядерный вариант.

Правда, пользователям PC'шек (а мы говорим в основном о них) от этого было бы ни холодно, ни жарко. Однако здесь «камнестроители» не заставили себя ждать: и AMD, и Intel очень быстро анонсировали, а затем и воплотили в реальность свои двухядерные решения, стоимость которых вполне вписывалась в рамки «суперкомпьютера для народа». По крайней мере, в лице лучших его представителей.

Так что пользователи оказались перед выбором между традиционными одноядерными процессорами с большей тактовой частотой или процессорами двухядерными – с меньшей (если оставаться в рамках одного бюджета). Как я уже говорил, рост тактовых частот уперся в потолок целесообразности: сколь бы велик он ни был (а тут имелся еще и потолок технологический), адекватного прироста производительности он уже за собой не влек. Но могли ли пользователи рассчитывать на хоть какой-то выигрыш в производительности от многоядерности?

Софтверная перспектива

Исходя из общих соображений, было очевидно, что ожидать двукратного увеличения быстродействия от самого факта удвоения числа процессоров (или их ядер) не приходится. Во-первых, на «железном» уровне два и более процессоров будут совместно использовать какие-то общие ресурсы компьютера – память, шины и так далее.

Во-вторых, неизбежны были потери быстродействия за счет «накладных расходов» – согласования операций, выполняемых на разных процессорах.

В-третьих, системные и прикладные задачи, выполняемые на многопроцессорной машине, должны допускать их распараллеливание, иначе любое увеличение количества «числодробителей» доставит мало радости пользователю.

И, наконец, в-четвертых, эффективность многопроцессорных конфигураций в значительной мере определялась стилем

выполнения пользовательских задач. Очевидно, что преимущественно интерактивные методы работы от удвоения «каменей» выиграют весьма мало – в любом случае тут узким местом окажется пресловутый человеческий фактор.

Решение проблем многозадачности на «железном» уровне было задачей производителей аппаратуры. А вот минимизация же «накладных расходов» и распараллеливание задач относились уже к сфере разработчиков ПО, в первую очередь – системного. Хотя в последнем случае роль создателей программ прикладных ничуть не меньше. Ну, а эффективное использование достижений тех и других – это уже вахта пользователей.

И нужно сказать, что пользователи UNIX-подобных операционных систем, в силу самой специфики их работы и укоренившихся привычек, оказались подготовленными к многозадачности лучше других. И были способны получить от нее больший выигрыш.

Ведь что происходит на типичной пользовательской UNIX? На ней постоянно что-то компилируется, архивируется и раз-

архивируется, кодируется и декодируется, бэкапится и восстанавливается. И все это – параллельно, и, в большей или меньшей степени, без интерактивного участия пользователя. Озаботивше-

гося, разумеется, заблаговременно, скриптами для запуска своих задач, выводом полученных данных в журналы-логи и прочие файлы, и так далее – за интерактивным режимом остается только просмотр результатов. И, конечно же, их обдумывание.

Так что вырисовывалась заманчивая картина: все это изобилие параллельно работающих задач выполнять действительно параллельно, раскидав по разным процессорам. Дело оставалось за малым – воплотить ее в кодах.

Изначально создатели UNIX (и ранних его клонов) ни о какой многопроцессорности не помышляли. И один из краеугольных камней его идеологии – концепция монолитных процессов, выполняемых на одном процессоре квазипараллельно, за счет квантования времени – казалось бы, препятствует реализации распараллеливания задач по разным «камням».

Тем не менее, когда многопроцессорные серверы и рабочие станции стали реальностью в индустриальном секторе, в дополнение к концепции процесса была создана и концепция т.н. нитей, или потоков [threads]. Это – части процесса, выполняемые параллельно и почти независимо друг от друга (в том числе и на отдельных процессорах), разделяющие, тем не менее, ресурсы составленного из них процесса. То есть собственного контекста, в том числе и отдельного пространства памяти, они не имеют, почему носят еще и имя легковесных процессов [light weight process] – обычные UNIX-процессы в этом случае можно называть «тяжелыми».

Само по себе понятие нитей возникло задолго до UNIX – чуть ли не со времен Очакова и ламповой электроники. И уже тогда были выявлены существенные недостатки этой концепции. Однако за истекшие годы ничего лучшего для поддержки мультипроцессорности придумано не было.

Как уже говорилось, проблема мультипроцессорности встала в первую очередь в индустриальном секторе. Где по ряду причин (в том числе и исторических) традиционно преобладали проприетарные представители UNIX-семейства. И разработчики последних доблестно эту проблему разрешили. Можно спорить, где она была решена лучше, где – не так хорошо, однако общепризнанно: масштабируемость многие годы был главной отличительной чертой (и главным козырем) и AIX от IBM, и Solaris от Sun, и прочих их братьев-конкурентов.

Свободные UNIX-совместимые ОС, как мы помним по первой статье цикла, разрабатывались преимущественно или в универ-

«Многопроцессорные серверы и станции стали реальностью.»

► Автор «стрекозы»
Мэтт Диллон.



ситетско-академической среде, или просто энтузиастами-любителями – как правило, на подручном оборудовании. Среди которого многопроцессорные суперкомпьютеры встречались не так уж и часто (солнце народной мультипроцессорности еще не показало из-за горизонта своих первых лучей). И потому поддержка многопроцессорности долго была слабой стороной и Linux, и BSD-систем (по крайней мере, для платформы Intel и совместимых).

Движение свободных операционок в корпоративный сектор, в первую очередь в качестве серверов разного рода, поставило перед ними задачи многопроцессорной поддержки и масштабируемости. И задачи эти постепенно решались: в том или ином виде многопроцессорные конфигурации давно поддерживаются ядром Linux и FreeBSD, затем такая поддержка появилась в NetBSD и OpenBSD. Тем не менее, ни одна из этих ОС не дотягивала еще до масштабируемости проприетарных UNIX'ов.

Правда, в мире свободного ПО испокон веков развивалось другое течение, косвенно связанное с многопроцессорностью – так называемые микроядерные ОС. Идея их в том, чтобы максимально возможную часть обязанностей ядра (взаимодействие с устройствами, файловыми системами и т.д.) вынести в пользовательское пространство памяти, оставив за ядром только коммуникационные функции. Теоретически рассуждая, это должно было бы обеспечить упрощение устройства системы, легкость ее портирования на новые архитектуры (в том числе и многопроцессорные), а также возможности масштабирования.

Из микроядерных решений наибольшее признание получило ядро Mach, разработанное в Университете Карнеги-Меллона, а затем развивавшееся в Университете Юты. Оно легло в основу ряда проектов разработки свободных ОС – самым известным из них долгое время был Hurd, разработка которого затем была переведена на другое микроядро – L4. Что, однако, не приблизило проект к состоянию, пригодному для применения. Однако существовали и другие попытки создания свободных микроядерных ОС – проекты Mach и Yami. И оба они свое развитие прекратили.

Таким образом, судьба свободных микроядерных проектов была (и остается) печальной. Что, помимо их невостребованности, объясняется еще и технологическими причинами: судя по всему, разработчики не смогли обеспечить приемлемую производительность своих систем, ведь упрощение устройства ядра влечет за собой усложнение межпроцессных коммуникаций.

Как ни странно, удачные реализации микроядерной архитектуры имели место быть в проприетарном секторе: на ранних версиях Mach основывалась знаменитая система NEXTStep, видевшаяся лет 15 назад платформой фантастического будущего. А предпоследняя, третья версия Mach легла, вместе с системными службами FreeBSD, в фундамент современной MacOS X.

Таков был исторический фон, на котором развернулись описанные далее события.

У истоков новой системы

Итак, где-то в середине июня 2003 г. Мэтт Диллон (Matt Dillon) вместе с группой товарищей сообщил о начале работы над новой ОС BSD-семейства – DragonFlyBSD. Возник сайт проекта – <http://www.dragonflybsd.org> – и репозиторий ее исходников, созданный 16 июня 2003 г. (этот день можно считать именинами системы) на кодовой базе FreeBSD четвертой ветки, имевшей статус стабильной (хотя в то время уже вовсю развивалась ветка пятая, вбирающая в себя все инновации BSD-мира).

Новая система получила и собственный тотем – стрекозу, что должно символизировать легкость и быстроту ее. Кстати, разработчики не гнушаются и сокращенных названий своей системы – DragonFly и даже DFBSD, к которым, вслед за ними, время от времени будем прибегать и мы.

Может возникнуть (и многократно возникает) вопрос: для чего нужна еще одна BSD-система? Разве не вдоволь насмотрелись мы на изобилие Linux-дистрибутивов, чтобы и BSD-системам желать той же участи?

Вопрос этот, конечно, носит сугубо риторический характер. Ведь если новые операционные системы создаются – значит, это кому-то нужно. И каждая такая система (если она, конечно, действительно нова и оригинальна) привносит в наш мир что-то свое, увеличивая, тем самым, сложность его и разнообразие.

А в оригинальности DragonFlyBSD отказать невозможно. Ибо, при практически полном внешнем сходстве с прототипом (FreeBSD 4.X), «внутри» у нее все было другое: управление памятью и процессами, представление о драйверах устройств и виртуальной файловой системе, вплоть до нового типа файлов – вариантных символических ссылок (varsims).

В основу DragonFly была положена модель легковесных нитей ядра [LWKT – Light Weight Kernel Thread]), что само по себе и не ново. Новым стал механизм планирования нитей – вместо единого планировщика [scheduler] их было введено несколько, по числу процессоров. Нити привязаны к своим процессорам изначально, однако допускается передача выполнения нити с одного процессора на другой при некоторых особых условиях. Данные отдельных нитей могут быть кэшированы независимо для каждого процессора.

Подобно системам с микроядерной архитектурой, в DragonFly максимум функций ядра вынесен из его пространства памяти в пользовательское пространство [userland]. В первую голову это относится к драйверам устройств – таким образом достигается рост как производительности, так и надежности системы, резко уменьшая вероятность ее «падения» под воздействием неправильно работающего драйвера.

Это повлекло за собой отказ от традиционного для UNIX механизма системных вызовов (каковой лишь эмулируется в целях совместимости). Его место занял механизм сообщений [messages] и их очередей, т.н. портов [ports], подобный применяющемуся в микроядре Mach, упоминавшемся выше.

При этом DragonFly не является микроядерной ОС – базовые функции по-прежнему возлагаются на ядро (и размещаются в его пространстве памяти). Однако почти все прочее может быть безболезненно собрано в качестве модулей юзерланда.

Таким образом, можно видеть, что основные инновации DragonFly ориентированы на работу в многопроцессорных системах. А вопрос о том, есть ли что делать простому пользователю на многопроцессорном, с позволения сказать, компьютере, мы рассмотрели уже в ходе обзора аппаратной ретроспективы.

Далее, важно, что если матушка DragonFly, FreeBSD, изначально предназначенная только для архитектуры i386, все более эволюционировала в сторону кроссплатформенности (в 5-й ветке к поддержке древней Alpha был добавлен SPARC, а затем и PowerPC), то наша «стрекоза» возвращается на исходные рубежи. И единственной поддерживаемой архитектурой в ней являет-

ся Intel-совместимая – на тот момент только 32-битная (64-разрядный вариант долго находился в состоянии разработки).

Такое ограничение в плане поддерживаемого «железа» может показаться отступлением от истинного UNIX Way. Однако на момент выхода DFBSD сбылось мрачное пророчество, высказанное 10 лет назад в одном компьютерном журнале:

«Через десять лет все платформы, кроме IBM PC, уйдут в небытие»

И все остальные архитектуры в качестве настольных платформ полностью утратили актуальность. Разработчики DragonFly считались с этой реальностью: в их тогдашних планах переноса на другие архитектуры не было (нет его и сейчас). Что компенсировалось возможностью оптимизации под платформу, единственно значимую практически. Это дало свои плоды – по визуальному быстродействию в настольных условиях DragonFly со дня своего зарождения существенно опережала FreeBSD как 5-й, так и 4-й ветки.

Наконец, в DragonFly на уровне ядра поддерживался механизм, напоминающий *prelinking* (предварительное связывание с разделяемыми библиотеками) – насколько мне известно, особенность почти уникальная и обещавшая значительный прирост скорости загрузки (а возможно, и быстроты исполнения) сложных программ, связываемых со множеством библиотек.

Все сказанное выше было технологическим обоснованием для того, чтобы отнестись к DragonFly не просто как к еще одному BSD-клону. Но это подкреплялось и субъективным фактором – личностью организатора проекта.

К моменту начала работы над DragonFly Мэтт Диллон был широко известен (в узких кругах) благодаря трем разработкам: Си-компилятору для платформы Amiga (именно из этой ОС пришла в DragonFly идея «ядерного прелинkingа»), утилите *dcron* и, главное, системе управления виртуальной памятью во FreeBSD. Не то чтобы он был единственным автором последней, однако вклад его в эту тему был одним из определяющих современной облик FreeBSD. Да и к аналогичной подсистеме ядра Linux он приложил руку.

Что немаловажно, в специальной статье (присутствующей в официальной документации FreeBSD) Мэтт сумел описать архитектуру виртуальной памяти языком, понятным для широких масс трудящихся. Очень рекомендую к прочтению – во введении к ней высказано немало интересных мыслей общего характера. Тем более, что она доступна и в русском переводе. А пока позволю себе процитировать ее фрагмент:

«Самой большой ошибкой, которую может допустить программист, является игнорирование истории, и это именно та ошибка, которую сделали многие другие современные операционные системы... Я плохо переношу тех, кого не учит история».

И время показало, что в DragonFly учли ошибки прошлого.

Некоторое время проект развивался как бы закулисно. Конечно, все желающие ознакомятся с прототипом системы могли свободно получить ее исходники с сайта проекта через CVS и развлекаться с ними в свое удовольствие (нужно ли говорить, что DragonFly распространялась и распространяется на условиях лицензии BSD?). Однако в виде, пригодном для установки простыми смертными, она не существовала.

Так продолжалось до мая 2004 года, когда один за другим начали появляться ISO-образы CD бета-версий DragonFly. Они не имели еще инсталлятора – следовало, руководствуясь документацией (вполне, впрочем, ясной, хоть и англоязычной), вручную разметить диск, создать файловые системы, перенести на них с дистрибутивного CD необходимые каталоги и произвести еще кое-какие манипуляции (типа создания файлов устройств и настройки

стартовых сервисов). Задача была не то чтобы сверхъестественно сложной, но и не вполне тривиальной.

А затем... Затем, в июне 2004 г., появился пре-релиз DragonFly, точнее, DragonFlyBSD 1.0RC1. От своих бета-предшественников он отличался тем, что уже имел инсталлятор – *BSD Installer*, разработанный в рамках самостоятельного проекта как универсальный установщик для любых BSD-систем. И впервые опробованный именно на DragonFly.

Надо заметить, что уже в те далекие времена (в масштабах истории DragonFly) установка этой ОС проходила без малейших осложнений (в том числе и на «железе», на которое FreeBSD устанавливалась с трудом). Однако к использованию система была пригодной лишь условно. Ибо, кроме базиса (соответствующего FreeBSD Distributions), не содержала почти ничего – ни Иксов, ни прекомпилированных пакетов (за исключением нескольких консольных утилит типа *cvsup* и *cdrtools*), ни собственной системы пакетного менеджмента.

Нужно сказать, что пре-релизная стадия для DragonFly оказалась очень короткой: уже 11 июля 2004 г. было объявлено о выходе релиза – DragonFlyBSD 1.0-RELEASE. Правда, и он просуществовал недолго: как это нередко бывает, в него вкралось несколько

мелких, но весьма неприятных ошибок, которые были выявлены мгновенно и столь же быстро исправлены в корректирующем релизе 1.0A.

Начиная с этого момента, DragonFly можно было считать

более-менее пригодной к использованию. Сам по себе дистрибутив по-прежнему не включал ни пакетов, выходящих за рамки базовой системы, ни системы портов. Однако с самого момента выхода релиза прекомпилированные пакеты для DragonFly можно было найти на двух самостоятельных сайтах.

При этом жизнь не стояла на месте, и дальнейшая работа над системой не прекращалась: с интервалом в 3–5 дней на FTP-сервере проекта и его зеркалах появлялись текущие снапшоты, одни из которых позиционировались как рабочие, другие же – как экспериментальные.

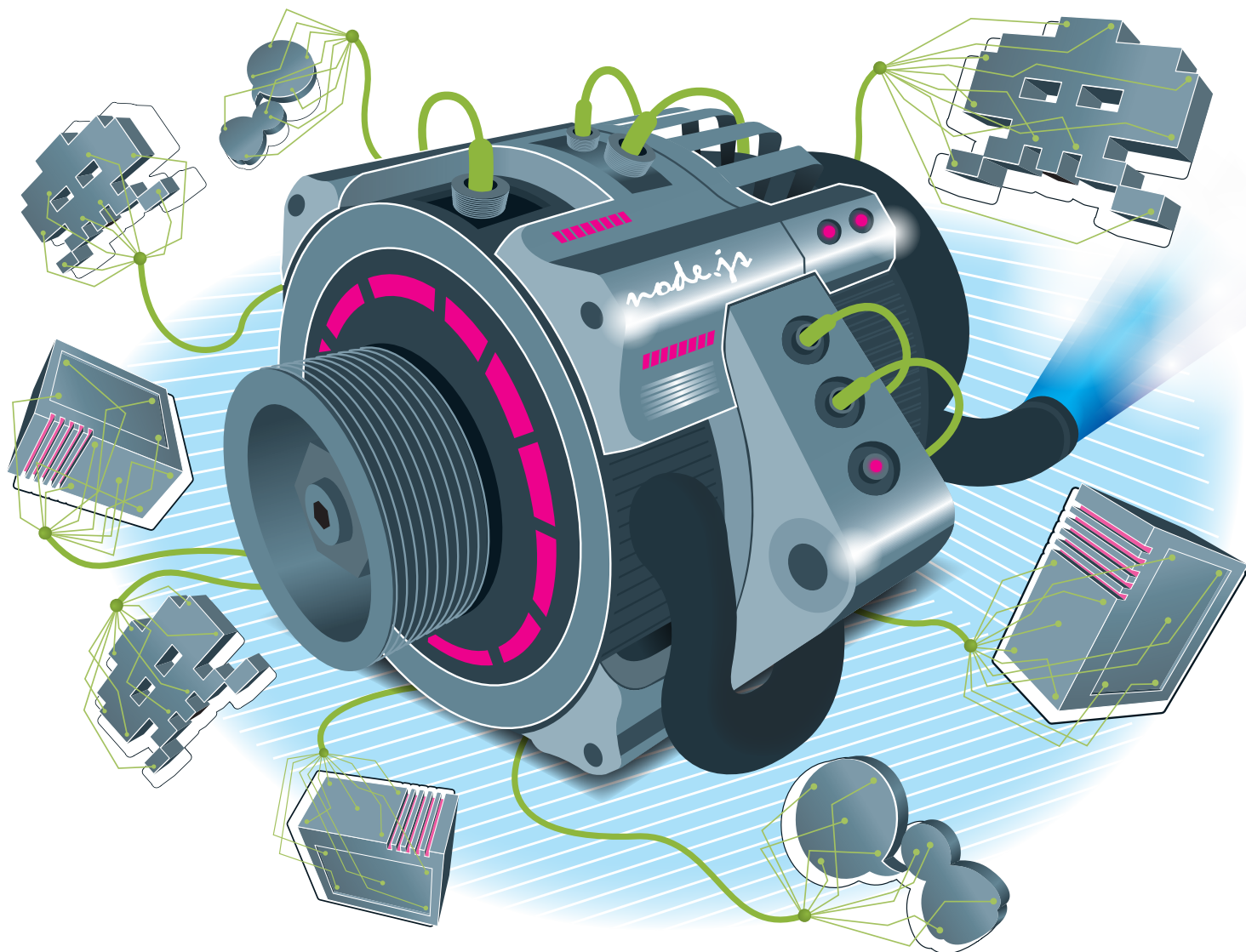
Двигалось дело и с портированием сторонних приложений. Первоначально оно осуществлялось с помощью адаптированной системы портов FreeBSD. Однако позднее разработчики пошли другим путем: прикрутили к DragonFly *pkgsrc* – кроссплатформенную портообразную систему, заимствованную из проекта NetBSD. И таким образом в распоряжении пользователя новой ОС сразу оказалось все изобилие открытого и бесплатного ПО, портированного на эту платформу – а надо отметить, что на NetBSD оно портировано в очень значительной своей части.

Первоначально предполагалось, что оба варианта – не более чем временные паллиативы, и в рамках проекта DragonFly будет разработана собственная система управления пакетами. Однако затем эта система была оставлена.

Такова была ранняя, короткая, но насыщенная событиями история операционной системы DragonFlyBSD на момент годовщины ее первого релиза. В последующие годы в ней появилось немало новшеств, как косметических (например, настоящая графическая консоль – аналог режима *frame buffer* в Linux), так и весьма кардинальных (собственная реализация виртуальной файловой системы, обеспечивающая доступ к практически всем файловым системам UNIX-подобных ОС, оригинальная файловая система Hammer). Статическое создание файлов устройств сменилось динамической файловой системой устройств *devfs*. И, наконец, DragonFly была портирована на архитектуру *x86_64*.

Однако предпосылки всего этого были заложены в первый год существования системы. Ныне ее история не окончена – но выходит за рамки рассматриваемого периода. **LXF**

«Время показало, что в DragonFly учли ошибки прошлого.»



Что за штука...

Node.js?

Джонатан Робертс наблюдает, как ведет себя JavaScript вне среды обитания.

В Значит, Node.js? А раз .js, то не обошлось без JavaScript? И, должно быть, не без web-браузеров, ведь JavaScript применяется там.

О И да, и нет. Это действительно JavaScript, но к браузерам не имеет никакого отношения.

В Совсем? Но разве JavaScript — это не язык программирования, благодаря которому мы открываем Google Maps, Aviary и Cloud9 IDE? Что же это, если не язык браузеров?

О Вы правы, это язык Сети, но применение JavaScript не исчерпывается созданием программ, запускаемых в браузере. Это также великолепный язык программирования общего назначения. По сути, JavaScript — два в одном. Есть ядро языка, включающее синтаксис и основные способы работы с текстом, массивы, даты и регулярные выражения — но не охватывающее многие другие функции, в том числе ввод/вывод, работу с сетью, хранение или графику. Все это обеспечивается «средой хоста».

В А хостом, разумеется, тут служит web-браузер?

О Именно так, вот потому-то все и думают, что это язык исключительно для web-программирования: в другой среде он им просто никогда не попадался. Вы можете считать, что Node.js представляет собой альтернативную среду для применения JavaScript. Речь идет о библиотеках и функциях, предусмотренных для ситуаций, отличающихся от тех, что могут встретиться в web-браузере.

В Хорошо, с этим ясно. Но какие именно задачи может решать JavaScript на Node.js?

О Он призван облегчить создание web-приложений — а точнее, серверных приложений. И это очень изящно, поскольку означает, что разработчики смогут создать и клиентскую сторону (то, что работает в браузере), и серверную, используя один и тот же язык. Тут есть потенциал для упрощения процесса разработки.

В Вроде и неплохо, но JavaScript — это же медленный, интерпретируемый язык? А web-серверы должны быть очень быстрыми, чтобы успевать реагировать на миллионы запросов.

О JavaScript — интерпретируемый язык, это верно, но не обязательно медленный. По факту, результаты некоторых (первичных) тестов на производительность показывают, что Node.js, в некоторых случаях, даже быстрее Nginx, а это высоко оптимизированный сервер на C.

В Хмм... что-то сомнительно. Не объясните ли вы, как такое возможно?

О Конечно. Причин тому две. Во-первых, роль JavaScript в Интернет становится все важнее — тому же Gmail без него просто не жить. И из-за широкой его распространенности, дабы пользоваться Интернетом было максимально приятно, разработчики браузеров много трудились над созданием «движков», сильно ускоряющих выполнение кода JavaScript.

В Ах, да, я что-то слышал об этом, когда Google запускал Chrome. Туда включили какой-то V8.

О Да, верно. V8 — это разработанный Google движок для JavaScript, и одно время он считался самым быстрым из всех существующих. Он действительно вывел скорость JavaScript на новый уровень. И самое классное в V8 то, что это открытый код, поэтому любой мог его взять и использовать в разнообразных приложениях, вне web-браузера.

В Стало быть, благодаря этому и появился Node.js?

О Именно так. Одна из важнейших частей Node.js — скоростной движок для JavaScript. И это первая причина, по которой он запускает программы на JavaScript так быстро. Но одно это не позволило бы ему тягаться в скорости с приложением, написанным на компилируемом языке типа C.

В И что же там еще накрутили, чтобы ускорить Node.js?

О Дело в том, что современные компьютеры и память у них мощные, а вот скорость записи, хоть с диска, хоть из Интернета, хромает. Причем настолько, что зачастую тормозит работу всего приложения. Значит, чтобы улучшить последнее,

мало просто разработать более быстрые алгоритмы, необходимо также оптимизировать и функции ввода/вывода.

В А такое вообще возможно в ПО? Конечно, мощность оборудования на это влияет, но улучшение самих программ ведь ничего не даст?

О Действительно, Node.js не оптимизирует ввод/вывод как таковой, зато не позволяет ему блокировать работу остальной программы. Дэн Йорк [Dan York] привел блестящую аналогию

«Одна из важнейших частей Node.js — скоростной движок JavaScript.»

(<http://tinyurl.com/49w15lk>) для этой модели. Представьте, что вы пришли к доктору. Отстояв очередь в регистратуру, вы узнаете, что для посещения врача требуется заполнить несколько анкет. Дальше возможно два сценария, первый: вы делаете это прямо там, не отходя от окошка. И пока вы заполняете, вся остальная очередь стоит и ждет — а сотрудник бездельничает.

В Кошмар какой! Да ни в одной приемной вы такого не увидите: будут либо километровые очереди, либо придется нанимать кучу регистраторов!

О Вот именно, но сегодня большинство сетевых приложения работает, грубо говоря, по такой модели.

В Хорошо, так чем в этой ситуации поможет Node.js?

О Ну, представьте опять же вашего доктора. Регистратор дает вам папку, ручку и анкету и отпускает заполнять ее, где вам удобно. Пока вы это делаете, он обслуживает других клиентов. Вы закончили — и снова встаете в очередь к окошку. Node.js делает то же самое.

Все медленные, сложные задачи (наподобие чтения файла с диска) он перенаправляет куда-нибудь еще, позволяя основному процессу Node.js обрабатывать другие запросы. Когда же завершится тот самый медленный процесс, он повторно ставится в очередь обработки.

В Звучит умно, но все же несколько запутано — переадресация, отслеживание всех этих задач...

О Да, не все так просто. В большинстве операционных систем для такого рода задач есть своя встроенная поддержка, но пользоваться ею трудно. А Node.js, на самом деле, все эти переадресации и отслеживания выполняет незаметно для глаза, представляя их в форме традиционных для Java «слушателей событий» [event

listeners] и «функций обратного вызова» [callback functions].

В Вот как? Да это уж само по себе звучит замысловато!

О Пожалуй, настолько замысловато, что нам здесь и места для разъяснений не хватит, но в терминах программирования все довольно понятно. А главное, они являются важнейшей частью современного JavaScript в сети, так что большинство разработчиков уже будет иметь четкое представление, как с ними работать.

В Ладно, поверю вам на слово. Итак, подводя черту... зачем же нужен Node.js?

О Он облегчает работу сетевых приложений — таких как web-серверы, чаты, онлайн-игры — все, что подразумевает одновременное обслуживание многих пользователей...

В ...за счет того, что не дает вводу/выводу тормозить работу всего приложения?

О Да, именно так! И не забывайте, что колесо скоро приложения Node.js написаны на JavaScript, такой подход к разработке будет естественным для всех, кому приходилось когда-либо создавать интерактивный сайт. Плюс, на JavaScript теперь может быть написано все приложение: как серверная, так и клиентская часть.

В Хорошо, я понял. А не приведете ли примеры каких-нибудь «сетевых приложений», где Node.js уже используется?

О Да их полно! Одна компания, под названием Voxel, разработала приложение, превращающее ваш смартфон в переносную рацию; другая создала игру Word² — по сути Скрэбл, но на огромной общей доске; третья представила дизайнерам онлайн-инструмент для каркасного моделирования, *Mockingbird*. Если хотите узнать о Node.js побольше или увидеть другие примеры его применения, загляните на сайт <http://nodejs.org>.

В Спасибо, непременно! И все же последний вопрос. Это самый Node.js, с его событиями и все такое, звучит здорово. Но многие разработчики не любят и не знают JavaScript. Им-то как быть?

О Очевидно, использовать Node.js они не смогут, поскольку он ориентирован исключительно на JavaScript. Но есть же много других библиотек и систем, работающих по тому же принципу: нравится Python — берите Twisted; предпочитаете Ruby — можете попробовать EventMachine; а если вы поклонник C — есть libevent; выбор за вами! **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.

Войны Linux

В конце 1980-х и начале 1990-х мы наблюдали бурное размножение версий UNIX, поскольку крупные разработчики вроде SUN, HP и IBM боролись каждый за свой стандарт для UNIX. Тогда многие думали, что эти так называемые «войны Unix» не способствуют проникновению UNIX на рынок.

Ясно, куда я клоню? История повторяется с Linux. Меня беспокоит не столько количество дистрибутивов, сколько разнообразие технологий внутри них. Например, в серверных продуктах мы видим две конкурирующие системы «мандатного контроля доступа» – SELinux (применяемую в RedHat) и AppArmor (Ubuntu и SUSE). На рабочем столе мы наблюдаем продолжительный раскол KDE/Gnome, а на подходе раскол X Server/Wayland. Даже в таких фундаментальных вещах, как загрузка системы, мы наблюдаем отказ от традиционного *init* ради замен типа *Upstart* (используется в Ubuntu и последней версии RedHat) и *systemd* (в последних релизах Fedora и OpenSUSE). Также есть разделение в управлении пакетами в стиле RedHat (RPM и *Yum*) и в стиле Debian (*dpkg* и *apt-get*). Короче, вы поняли идею.

Фрагментированный рынок

Имеет ли это значение? Ну, это, естественно, сильно усложняет жизнь системным администраторам, поддерживающим несколько дистрибутивов. Фрагментируется и рынок систем обучения. А он, поверьте, достаточно узок и без фрагментации. Ныне почти невозможно преподавать системное администрирование Linux, абстрагировавшись от компаний-производителей. И этот огромный набор технологий широко и гораздо менее определенным способом мешает внедрению Linux. И у меня нет решения для этой проблемы. Linux – открытая система, и я не хотел бы этого менять. Но именно открытость делает возможными «войны Linux». Ваши мнения? Напишите мне пару строк на chris.linuxformat@gmail.com.

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



Клички для команд

Вы когда-нибудь думали о том, почему **cat** называется **cat**? Прольем немного света на некоторые еще более темные имена команд.

Как-то я рассказал жене о студентах своих курсов, с трудом запоминающих имена команд. Моя жена рассудила, что им полегчало бы, будь все команды внятными английскими словами. Здесь есть доля истины: ведь любая система, где действует команда **shopt dotglob**, требует пояснений. В улучшенном моей женой Linux классические двухбуквенные команды вроде **ls**, **cp** и **mv** расширились до **list**, **copy** и **move**. Но я, честно говоря, счел это плохой идеей. Если применить английские слова, французы тут же захотят перевести их на свой язык – получатся **dénombrer**, **copier** и **déplacer**, и нашим скриптам какую.

Все же некоторые имена команд, с которыми мы сталкиваемся в Linux, и в самом деле загадочны. Вот объяснения для нескольких из них:

- » **cat**: Сокращение от «catenate», означающего «соединить в цепочку» (то же, что и «concatenate”).
- » **grep**: Это название пришло из старого редактора командной строки, **ed**, в котором эквивалентная команда имела вид **g/re/p**, где **re** – регулярное выражение, по которому производится поиск. Эта команда – в большей мере, чем любая другая – вошла в общепотребительную речь как глагол: «Why don't you grep for it in the config file [А не грепнуть ли это в файле настройки]?»
- » **crontab**: Демон **cron** (запускающий задания по расписанию) назван в честь Хроноса, древнегреческого бога времени. А **crontab** – это просто «Cron Table» (Таблица Cron).
- » **bash**: Сокращение от «Bourne Again Shell» (Снова оболочка от Борна) – игра слов, ссылка на старую оболочку от Стива Борна [Steve Bourne].

- » **dd**: Говорят, что эта странная утилита «конвертации и копирования» получила имя из языка управления заданиями IBM, тайнописи для выражения просьб к ЭВМ перед тем, как она только подумает о запуске вашего задания. **DD** – сокращение от «data description» (описание данных).
- » **less**: Ссылка на более старую программу «more», с помощью которой можно просматривать текстовый файл по строкам. Команда **more** прокручивает текст только вперед, **less** умеет и назад.
- » **vi**: Сокращение от «visual» (визуальный), но изначально – режим редактора **ex**. Произносится как «ви-ай», а не как «ви» и уж точно не как «шесть». Может, этот редактор и стар, но не прибегает к нумерации древних римлян.
- » **dig**: Утилита для ручного выполнения DNS-запросов (при тестировании). Говорят, что это сокращение от «domain information groper» (копатель информации о домене), но я не уверен.

Настоящие загадки

Этимология нескольких имен была полностью утрачена. Почему учетная запись суперпользователя называется **root**? Почему раскрытие маски называется **globbing**? Почему каталог с настройками системы называется **etc**, а привилегированная группа в системах UNIX называлась **wheel** [колесо]? Если у вас есть ответы на эти вопросы или вы знаете что-то интересное о происхождении других имен, напишите мне на chris.linuxformat@gmail.com.

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



Укращение облака

Cloudmin позволяет централизованно управлять виртуальными системами, по различным технологиям. Присоединяйтесь к нам – попробуем его в действии.

Широкое распространение виртуализации и облачных вычислений привело к большому скачку количества логически отдельных компьютеров, которыми нужно управлять. У хостинг-провайдеров и других крупномасштабных пользователей обычно оказывается смешанный набор систем: некоторые из них – виртуальные машины на офисных компьютерах, некоторые – на облаках EC2 или Rackspace, некоторые – даже старые добрые физические серверы. Не здорово ли было бы управлять всеми этими системами единообразно из одного административного интерфейса? Конечно!

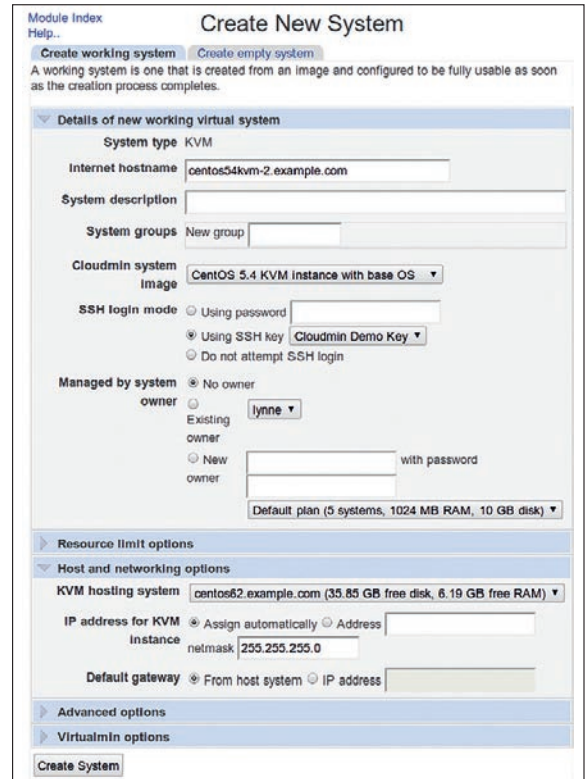
Cloudmin – одно из таких решений. Цитируя его сайт (<http://www.virtualmin.com/about-cloudmin>), «Cloudmin предоставляет web-интерфейс для управления множеством систем, обычно виртуальных, работающих на базе технологии виртуализации наподобие Xen или OpenVZ. С помощью одного Cloudmin можно управлять несколькими хостами, в каждом из которых, в свою очередь, может быть несколько виртуальных систем».

Cloudmin доступен в различных версиях. Есть бесплатная версия (Cloudmin GPL), с которой можно управлять только виртуальными машинами Xen или KVM в одной основной системе. GPL-версия также не включает управление системами на EC2 и управление не виртуальными (физическими) хостами. Затем есть версия Pro – в ней можно управлять несколькими хостами и несколькими технологиями виртуализации, в том числе Xen, OpenVZ (контейнерная технология, которую я описывал в прошлом месяце), KVM, Linux VServers, Solaris Zones и Amazon EC2.

Хотя Cloudmin Pro не бесплатен, он и не дорог. Сейчас у них проходит акция со скидкой 50 %, и годовую лицензию на 50 систем можно приобрести за 299 долларов, что в результате дает около 50 центов на систему в месяц. Хозяин Cloudmin Джейми Кэмерон [Jamie Cameron] любезно предоставил мне пробную лицензию на версию Pro, поэтому откиньтесь на спинку кресла и следите за моими экспериментами на тест-драйве.

Установка

Простейший способ начать работу с Cloudmin – запустить автоматизированный установочный скрипт, который с помощью стандартных команд управления пакетами устанавливает Cloudmin напрямую из программных репозиториях Cloudmin. Поэтому – как всегда, в поисках немедленного удовлетворения – я так и сделал. На сайте Cloudmin рекомендуется использовать Red Hat или CentOS; я начал с установки CentOS 6.2 на средний (4-ядер-



➤ Окна Cloudmin скорее функциональны, чем красивы. Здесь создается новая виртуальная машина KVM.

ный, 8 Гб) сервер. Я создал более или менее простую конфигурацию сервера (без рабочего стола), хотя и включил в нее утилиты виртуализации, и запустил yum update, чтобы обновить систему до актуального состояния.

Далее нужно было загрузить и запустить установочный скрипт Cloudmin. Этот скрипт привязан к лицензии Cloudmin, а ссылка на загрузку доступна после принятия лицензии. (Для GPL-версии зайдите на <http://www.virtualmin.com/documentation/cloudmin/gpl>, там вы найдете подробную информацию и ссылки на загрузку установочного скрипта). Скрипт автоматически устанавливает Cloudmin и все зависимости. Отработал он как часы: выполнялся минут пять, и процесс прошел без сучка, без задоринка. После установки сервер Cloudmin слушает порт 10000, поэтому пришлось открыть его в брандмауэре; затем я смог открыть адрес <https://centos62.example.com:10000>, войти в систему от имени суперпользователя-root и увидеть домашнюю страницу Cloudmin. Если установочный диск CentOS у вас наготове, то весь процесс займет, пожалуй, от 30 до 60 минут.

Однако далее кривая обучения довольно круто идет вверх – функций, которые можно исследовать, МНОГО, и надо прочесть массу документации. Я поставил себе, на мой взгляд, реалистичную цель: создание и управление несколькими виртуальными машинами на KVM из главной системы и несколькими компьютерами на EC2 из моей учетной записи Amazon Web Services.

Следующий шаг после запуска Cloudmin – зарегистрировать хост, которым я буду управлять из главной системы. Это было довольно просто, поскольку у меня всего один физический сер-

Скорая помощь

Есть хорошие руководства по Cloudmin, но простых ссылок на них с главной страницы я не нашел. Чтобы сэкономить время на поиски, заходите на www.virtualmin.com/documentation/cloudmin.

Знакомьтесь с родней

Cloudmin – близкий родственник двух других утилит администрирования через web-интерфейс, Webmin и Virtualmin. Webmin (о котором я рассказывал в LXF155) состоит из web-сервера и набора модулей расширения, позволяющих выполнять широкий набор действий по настройке системы через web-браузер, без работы с файлами в /etc, без командной строки и без изучения текстового редактора.

Virtualmin – панель управления для web-хостинга, которая также предоставляет средства управления виртуальными доменами, почтовыми ящиками, сайтами, базами данных и приложениями.

Все три программы очень похожи – вероятно, потому, что все они используют изрядный общий блок низкоуровневого кода на Perl.

Резервные копии

С помощью *Cloudmin* администратор главной системы и ее владельцы могут создавать резервные копии виртуальных систем для *Xen*, *OpenVZ*, *Vservers* или *Solaris Zones* (обратите внимание, что двух из выбранных для этого эксперимента технологий, *KVM* и *EC2*, в списке нет).

Резервное копирование можно выполнять вручную или регулярно по расписанию – скажем, ежедневно. Копия включает содержимое файловой системы виртуальной машины или образов диска либо сжатый архив виртуальных дисков (для *Xen*), либо *tar*-архив для других технологий. *Cloudmin* может сохранять резервные копии в различных местах – по *SCP*, *FTP* или по любой другой технологии, поддерживаемой *Cloudmin*.

вер, так что главная система (в которой работает ПО *Cloudmin*) и система хоста (где я буду создавать свои виртуальные машины) – это один и тот же компьютер. Но можно зарегистрировать и дополнительные физические хосты в количестве, разрешенном лицензией. Есть даже возможность просканировать сеть на наличие подходящих кандидатов. Следуя плану своей миссии, я зарегистрировал компьютер как систему хоста для *KVM* (его также можно зарегистрировать как хост для *Xen*, *Citrix Xen*, *VServer*, *Solaris Zones*, *OpenVZ* и *LXC*). Вам понадобится указать доменное имя для виртуальных машин на этом хосте, и можно задать диапазон IP-адресов, которые хост выделит виртуальным машинам при их создании. Если у вас несколько физических хостов, распределение IP-адресов между ними придется спланировать заранее.

Получение образа системы

Следующей задачей была загрузка образа системы. Образ системы *Cloudmin* – файл, содержащий все данные для создания новой виртуальной системы. Обычно это *tar*-архив или дампы файловой системы всей виртуальной ОС. При создании виртуальной системы образ используется для заполнения начальной файловой системы виртуальной ОС. Я загрузил несколько образов на основе *KVM* – один для *CentOS 5.4* и один для *Ubuntu 10.10*. Образы отличаются в размерах, от 200 МБ до 1 ГБ. На сайте *Cloudmin* приведен список из 68 образов, но это количество по сути втрое меньше, так как большинство образов доступны в трех вариантах: один только с базовой ОС, один с установленной *Virtualmin GPL* и один с установленной *Virtualmin Pro*. Каждый образ предназначен для конкретной технологии виртуализации. Я насчитал около 36 образов для *Xen* (или *Citrix/Xen*), шесть для *OpenVZ*, 11 для *KVM* и шесть для *Solaris Zones*. Если вы не нашли образ по вашим потребностям, в *Cloudmin* есть поддержка создания собственного образа, исходя из дампа диска (или основного раздела) существующей системы.

Наконец, я приступил к созданию виртуальной машины. Как вы видели на экранном снимке на предыдущей странице, здесь нужно ввести довольно-таки много параметров, включая выбор имени хоста, группы (о них поговорим позже), образа системы, логина и пароля *ssh* (или пары ключей), задать объем используемых оперативной памяти и дискового пространства, IP-адрес и т. д. После заполнения формы на создание системы

и установку в нее образа ушло около 70 секунд. Насколько я могу видеть, здесь нельзя «создать десять экземпляров системы», как допускается интерфейсом пользователя *web*-сервисов *Amazon*. Но всего несколькими щелчками мыши можно клонировать существующую систему. Клон получает новые имя хоста и IP-адрес, но в остальном является копией оригинала с точно такой же файловой системой, паролями, ограничениями ресурсов и владельцем. И я создал клон, просто ради развлечения.

На моей первой виртуальной машине вход в систему производился с помощью пароля, но для второй я решил создать пару ключей для *ssh*. С командной строки в главной системе я запустил *ssh-keygen* для генерации пары ключей, которую затем импортировал в *Cloudmin*. Связав пару ключей с системой, я смог войти в нее без пароля с компьютера, на котором находился закрытый ключ.

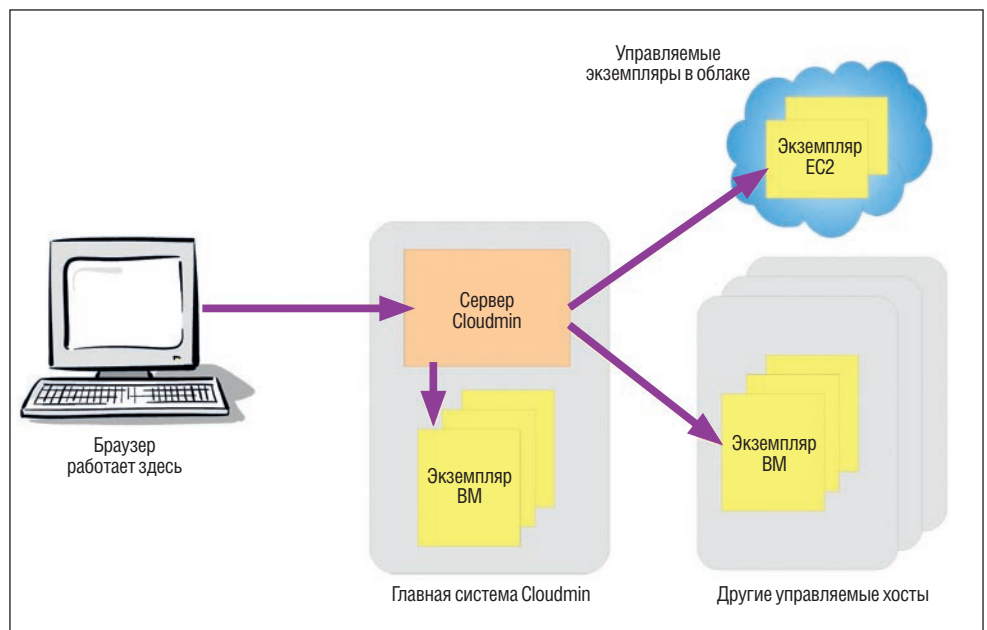
Пустая система

Также можно создать экземпляр просто «пустой системы», которая создается без первоначальной установки ОС на ее виртуальный диск. При этом требуется указать образ CD или физический привод CD-ROM, подключенные к системе и используемые для установки ОС. По завершении установки системой можно управлять так же, как и любой другой виртуальной машиной. В принципе, можно устанавливать и другие системы (не *Linux*), но возможности управления такими системами в *Cloudmin* ограничены.

Рассмотрим некоторые возможности управления системами, которые мы создали. Можно менять пароли на вход; клонировать систему или создать из нее образ и изготовить новые экземпляры системы; установить в нее *Virtualmin* или *Webmin*; переместить ее на другой хост (если у вас более одного хоста); откатить систему в ее первоначальное состояние, открыть терминал с правами *root* (в браузере); запускать в нем различные команды и передавать и принимать файлы, хотя пользовательский интерфейс для этих операций довольно аскетичен. Наконец, систему, разумеется, можно приостановить, выключить, перезагрузить или удалить целиком.

Пока я выполнил только половину своей миссии – создал три виртуальных машины *KVM* (*CentOS*, *Ubuntu* и один клон), и могу войти в систему на них от имени *root*. Вторая половина – создание и управление компьютерами на *EC2* – требует предварительного копирования и вставки для импорта данных моей учетной записи *AWS* в *Cloudmin*, чтобы программа пользовалась API управления

➤ Сервер *Cloudmin*, запущенный в главном хосте, предоставляет *web*-интерфейс для управления локальными и удаленными виртуальными машинами.



» Я добился своей цели — шесть управляемых систем! В этом окне отчасти возможно и «массовое управление системами».

Hostname	Type	OS	Status	Description
centos54kvm.example.com	KVM	CentOS Linux	Webmin - 168 updates	CentOS 5.4 KVM instance with base OS
centos62.example.com	Master	CentOS Linux	Webmin - 1 update	This system
ec2-107-20...zonaws.com	Amazon EC2	Fedora Linux	SSH	Cloudmin Demo Server 2
ec2-107-22...zonaws.com	Amazon EC2	Fedora Linux	Webmin down	Cloudmin Demo Server 1
ub2kvm.example.com	KVM	Ubuntu Linux	SSH	Clone of Ubuntu 10.10
ubuntu1010...xample.com	KVM	Ubuntu Linux	SSH	Ubuntu 10.10

AWS от моего имени. С ключами, должен признать, у меня плохо. Как-то раз я уехал из дома на неделю с ключами от машины жены в кармане. Но я аккуратно импортировал идентификатор учетной записи, идентификатор ключа доступа, секретный ключ доступа, сертификат и секретный ключ X.509 AWS. Это проще, чем кажется, если у вас есть учетная запись AWS, так как большая часть этой информации доступна в меню Security Credentials [Параметры безопасности] консоли управления AWS.

Если вы когда-либо создавали экземпляр компьютера с Linux для EC2, то знаете, что единственная поддерживаемая им форма аутентификации по ssh – аутентификация с помощью пары ключей. Парольная аутентификация отсутствует. Поэтому создайте пару ключей в EC2, загрузите их в главную систему *Cloudmin* и затем импортируйте в *Cloudmin*. С формой «Создать экземпляр EC2 [Create EC2 Instance]» я хлебнул горя: как и со многими вещами в жизни, в ней проще ошибиться, чем сделать все правильно. Осложнение вызывало то, что когда я возвращался в форму ради новой атаки, заполнять ее приходилось с самого начала. Это, кажется, вообще свойственно *Cloudmin* – данные, вводимые в формы, дотошно проверяются, но при повторе попытки правильная часть не перезаполняется.

Все же я это добил. На создание системы на EC2, ее загрузку и проверку ее состояния с помощью входа в нее по ssh ушло около двух минут. Вторую систему, разумеется, было создавать гораздо проще. В *Cloudmin* также можно импортировать существующие системы на EC2, но я этого не пробовал.

Наконец, у меня получилось шесть управляемых *Cloudmin* систем – главный хост, две локальных системы и клон одной из них, а также несколько экземпляров EC2. Вы видите их на экранном снимке вверху. Миссия завершена!

Итоги

Cloudmin – система с впечатляющими возможностями. Оглядываясь назад, я заметил, что за все время едва коснулся командной строки – лишь однажды, чтобы изменить настройки брандмауэра, и еще раз, чтобы сгенерировать пару ключей ssh. Пользо-

вательский интерфейс несколько неряшлив – немного графики и иконок его украсили бы; и в идеальном мире я хотел бы увидеть пару мастеров, которые помогли бы решить самые популярные задачи. Возможность программы осуществлять «массовое управление» ограничена: можно лишь ввести команды для запуска на нескольких системах, но не более того.

С учетом потенциально гетерогенной природы систем, которыми управляет *Cloudmin*, большего, пожалуй, ожидать и не приходится.

Некоторые возможности *Cloudmin* я толком не исследовал. Они включают:

» **Группы расположения:** При создании новой системы вас, вероятно, интересует физическое расположение ее экземпляра, но может быть все равно, на каком хосте она создается. Группа расположения – группа хостов, находящихся в одном ЦОД, которую можно выбрать при создании виртуальной системы. Сам конкретный хост затем выбирается из членов группы расположения с учетом объема свободной памяти, дискового пространства или случайным образом.

» **Группы отказов хоста:** Группа хостов, совместно использующих хранилище для виртуальных систем; поэтому они могут приходить на выручку друг другу, если один из хостов выйдет из строя. Это позволяет автоматически перезапустить виртуальную систему на другом хосте с теми же виртуальными дисками или файловой системой в случае отказа оборудования.

» **Системные предупреждения:** *Cloudmin* может отслеживать управляемые системы и выдавать предупреждения, если использование ресурсов оказывается ниже или выше указанного порога. Предупреждения можно задавать для отдельных систем, для всех систем или для систем конкретного типа, например *OpenVZ* или *KVM*. Отслеживаемые ресурсы – использование процессора, памяти и свободное дисковое пространство. При возникновении предупреждения *Cloudmin* может отправить письмо администратору, назначенным «владельцам» системы-нарушителя или на любой предусмотренный вами адрес.

» **Владельцы системы:** Владелец системы – ограниченная учетная запись *Cloudmin*. При создании системы ей можно назначить владельца; тот может войти в *Cloudmin*, но увидит только компьютеры, которыми он может управлять. У владельца есть административный доступ к своим системам, и он может перезагружать и даже удалять их, но не может выполнять операции в системе хоста или изменять глобальные настройки.

Можно также задать учетные записи с ограничением используемого дискового пространства, процессора, памяти и полосы пропускания сети. Можно отслеживать использование ресурсов и вычислять стоимость их оплаты. Это прекрасные возможности для хостинг-провайдеров, предоставляющих своим клиентам «облачные» сервисы. LXF

Взгляд из командной строки

Для пробы возможностей *Cloudmin* из командной строки (например, автоматического управления из скрипта) существует команда `cloudmin`; она дает доступ почти ко всем функциям, доступным через web-интерфейс. У нее есть более ста подкоманд с такими именами, как `clone-system`, `create-system` и `delete-system`. Если *Cloudmin* установлен, то команда

```
cloudmin --help
```

выведет их полный список.



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул PostgreSQL на защите кандидатской диссертации по физике элементарных частиц.

Kickstarter, игры и Linux

Переносимость нужна людям, не умеющим писать новые программы.
Линус Торвальдс

После былинного успеха *Double Fine Adventure* Тима Шейфера [Tim Schafer] (собрали больше \$3,3 млн, гарантирован порт под GNU/Linux) на Kickstarter (сайт организации народного финансирования творческих проектов) в секции игр забурлила жизнь. Идея, что пользователи могут дать больше, чем просили (проект FTL: Faster Than Light собрал \$200 тыс. вместо запрошенных 10 тыс.; тоже будет порт под GNU/Linux), взбудоражила умы игроделов.

На Kickstarter они идут не от хорошей жизни. Далеко не все проекты находят поддержку у издателей, ориентированных на прибыль. Дивный проект, о котором вы грезили с детства, пройдя «подставьте любимую игру типа *Fallout*, *Planescape*, *Dungeon Keeper*, *Chrono Trigger* и т.д.», в нынешнем мире сильно уступает по показателю прибыли «Злобным Птицам»: героев делает рекламный бюджет, и пошаговым RPG нет места.

Что же делать? Как вариант – давать деньги в обход издателей. И тут пользователи GNU/Linux имеют шанс быть услышанными: это только в мире «Злобных Птиц» их меньше 2%, а в мире народного финансирования их финансовая доля, по статистике Humble Indie Bundle, свыше 20%. Этим нужно пользоваться, намекая разработчикам на то, что они приобретут, сделав порт под GNU/Linux.

P.S. *Trine 2* вышел, в том числе и под GNU/Linux. Я в растерянности: сразу покупать или дожидаться акции HIB? E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Достигать эффектов ЛОМО 70
Ник Вейч наглядно демонстрирует, как из качественных фото изготавливаются изображения с иллюзией антикварной лomoграфии при перекрестной обработке в *GIMP*.



Кодировать видеофайлы 74
Заблудиться в видеоформатах так же легко, как в экваториальных джунглях – но только не с **Джонатаном Робертсом**. Он выведет вас на верную тропу.



Автоматически входить 78
Подключайтесь к своему компьютеру откуда угодно, не выставляя его беззащитным под атаку злоумышленников: безопасность обеспечивает **Нейл Ботвик**.



Продвигать Arduino вдале 80
Вы любите повозиться с Arduino, но считаете, что его датчики немного недалёковидны? **Ник Вейч** приставит к ним дистанцию радиосообщения.



Вникать в Erlang 84
Андрей Ушаков декодирует разнообразные данные, которые он в прошлый раз закодировал. Главное, чтобы данные получились такие же.



Строить базу данных ТВ 88
Примените MySQL для создания базы данных всех любезных вашему сердцу телесериалов, не без помощи **Дейва Кросса**.

Пропустили номер?

Узнайте на с. 104, как получить его прямо сейчас!





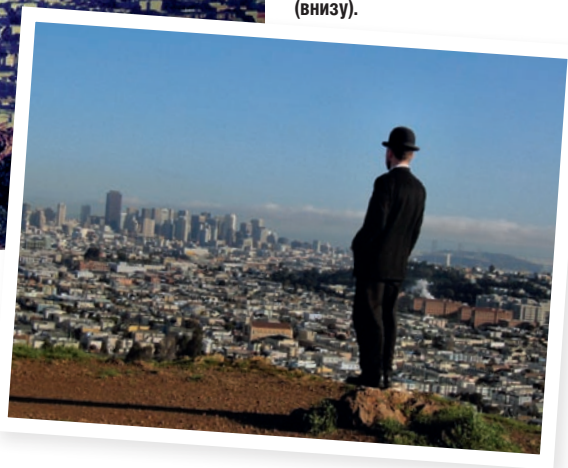
В стиле ЛОМО

Фотоэстет **Ник Вейч** показывает, как с помощью *GIMP* придать вашим дорогим снимкам такой вид, словно их сделали игрушечной камерой.



► После: наш отредактированный снимок (слева) имеет типичный вид перекрестной обработки фотографии ЛОМО, с особым акцентом на голубое небо.

► До: наш исходный снимок, сделанный цифровой камерой (внизу).



Наш эксперт

Когда **LXF** появился, его держали на плаву только скрипты *Bash* от **Ника Вейча**. Потом их заменили «люди», и это, по мнению Ника, стало шагом назад...

Как ни странно, в наши дни недорогой, замечательно точной и правдоподобной цифровой фотографии большее впечатление на людей часто производят снимки, сделанные на пленку игрушечной камерой с крошечными пластмассовыми линзами. Такой тип изображений часто называют ломографией [Lomographs], по названию популярного бренда пленочных камер, который за недавние годы претерпел обновление.

А если вам нравятся такие изображения, но вовсе не хочется возиться с пленкой, проявкой и долгим ожиданием готовых снимков – *GIMP* более чем способен взять обычную фотографию с вашей куда более качественной цифровой камеры и превратить ее в изображение перекрестной обработки, вокруг которых столько восторгов на Flickr.

Хитрость в том, чтобы имитировать эффекты пленочной камеры, внося небольшие изменения в изображение. Основная разница – в том, как пленка «видит» цвет, в отличие от датчика цифровой камеры.

За долгие годы на пленке применялись химические вещества с самой разной светочувствительностью; основная суть была в том, что они по-разному реагировали на разную длину световых волн. Цветочувствительность – это то, что придает цвету характерные теплые или холодные оттенки, в зависимости от способа обработки. Для подчеркивания этого эффекта энтузиасты час-

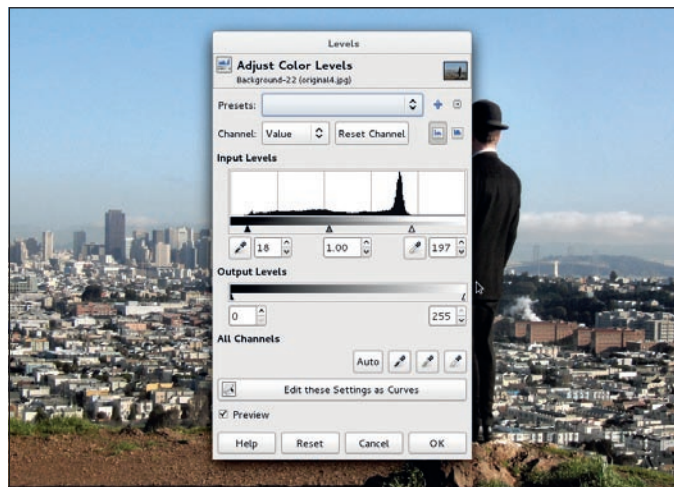
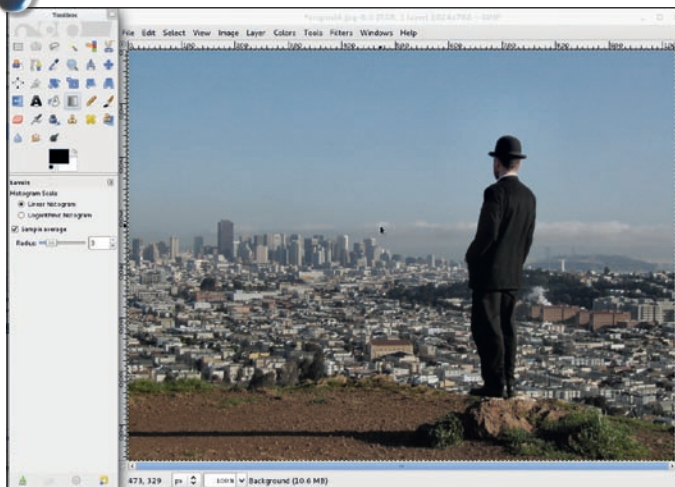
тенько подвергали пленку «перекрестной обработке» (взяв «неправильные» реагенты), но такое гораздо легче сделать в *GIMP*.

Однако одним из главных условий является выбор правильного изображения, это на первом месте. Эффекты ЛОМО преувеличивают голубой цвет, так что если вы возьмете пейзажный снимок, где много неба – это хорошо, и море тоже очень подойдет. Если подходящего снимка у вас нет, поищите его у Creative Commons. Для нашего урока мы позаимствовали изображение у striatic, пользователя Flickr (<http://www.flickr.com/photos/striatic/106570454/>).

С помощью этого руководства вы сможете, в погоне за «ЛОМО-подобием», делать с вашими изображениями что угодно. Популярно добавление искусственных световых бликов или эффекта перфорации пленки, для достижения полной иллюзии пленочного снимка. Но не забывайте делать все в разных слоях: тогда легко будет откатить неудачную попытку.



Шаг за шагом: Редактируем изображение

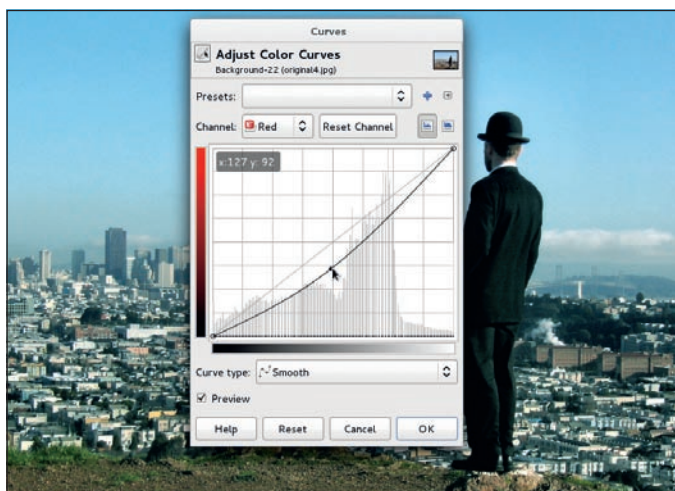


1 Выберите картинку

Правильный выбор стартового изображения кажется очевидностью, но в стиле ЛОМО лучше работают именно изображения определенного типа. Очень хорошо, если на картинке много голубого неба, потому что цветовой сдвиг придаст ему дополнительную энергию. Важно и наличие яркого освещения, иначе может оказаться слишком темно, чтобы понять, что же есть на фото. Интерьер или изображение с мелкими деталями не слишком хороши в работе, поскольку цветовой сдвиг и эффект контрастности замутнят некогда четкие детали и сделают их неразличимыми. Если у вас нет подходящего изображения, вы найдете их в большом количестве на сайтах под лицензией Creative Commons, таких, как www.flickr.com.

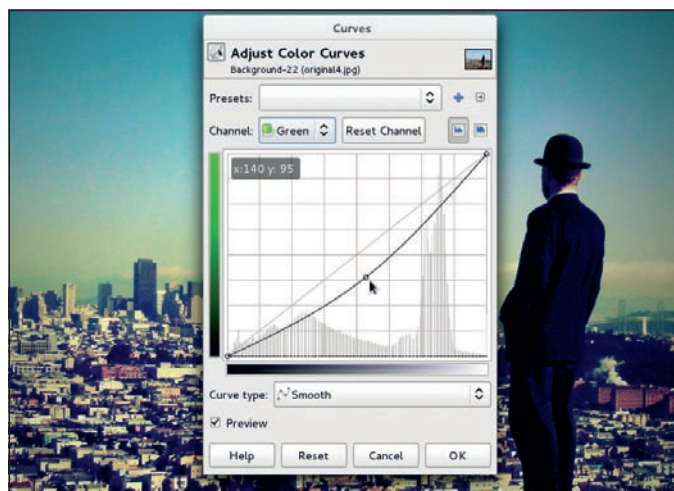
2 Подгонка изображения

Даже если выбрано довольно хорошо освещенное изображение, все же будет лучше настроить его общий диапазон. Откройте Colours > Levels [Цвета > Уровни] и используйте либо ползунок, чтобы опустить точку White [Белый] туда, где на гистограмме имеются линии, либо пипетку, чтобы выбрать ту часть изображения, которая намечена как самая яркая. Прделайте то же самое с другого конца диапазона, с параметрами Black [Черный], чтобы изображение стало более контрастным. Стрелка Grey [Серый] в середине гистограммы должна оставаться примерно посередине между ними, если только вам не нужно исправить недодержанный снимок.



3 Цветовая кривая: красный

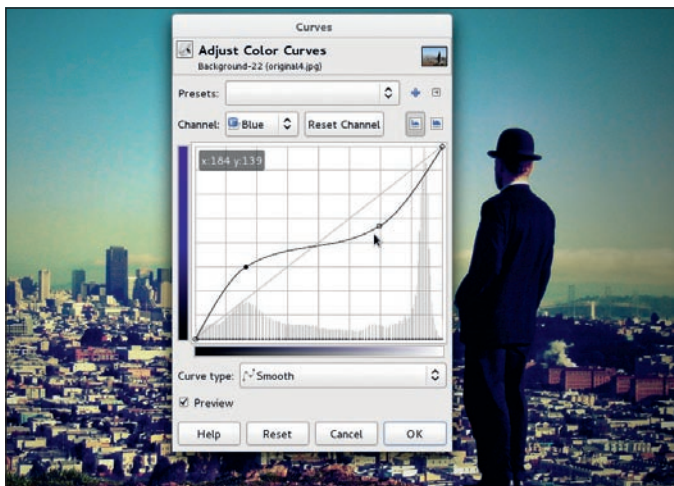
Нам нужно настроить цветовые кривые, чтобы учесть прохождение света через пластиковые линзы плохого качества на устаревшую пленку! Здесь используется своего рода цветовой сдвиг – изменение состава цветов RGB на изображении. Перейдите в Colours > Curves [Цвета > Кривые] и выберите канал Red [Красный]. Перетащите кривую слегка ниже линии, как показано. Но не переборщите! Для достижения эффекта перекрестной обработки нужно получить S-образную кривую, чтобы верхняя часть кривой вышла за линию. При необходимости вы всегда можете вернуться и настроить эти параметры.



4 Цветовая кривая: зеленый

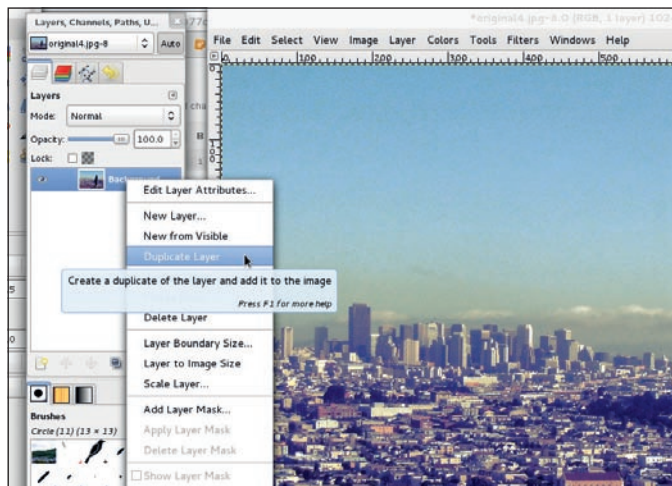
Теперь переключитесь на канал Green [Зеленый] и проделайте то же самое, потянув кривую ниже центральной линии за одну точку. Этим мы снизим общее количество зеленого и красного в полутонах изображения, не затронув параметры White (потому что верх кривой по-прежнему остается в той же точке). Не заходите слишком далеко, но изображение теперь должно иметь больше голубых оттенков. Так же, как и с кривой Red, если вы хотите создать более сильный эффект перекрестной обработки изображения, вам нужно сделать S-образную кривую. Сделать это намного проще с помощью двух точек, а не одной (просто щелкните по кривой и перетащите ее), но кривая должна быть гладкой, иначе цветовые изменения в отдельных пикселях будут слишком резкими.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



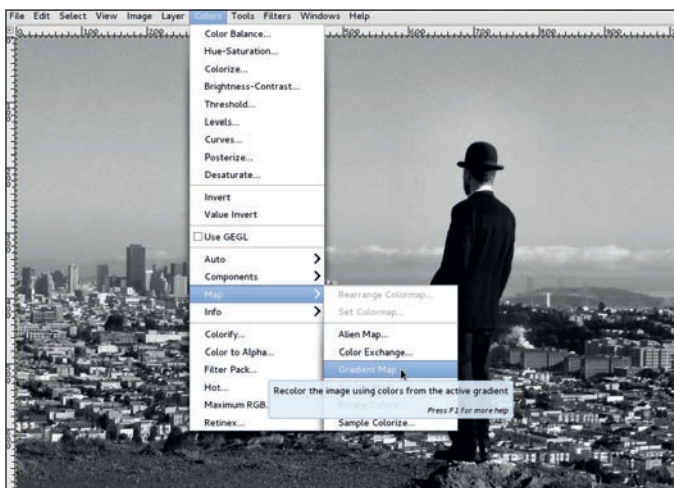
5 Цветовая кривая: голубой

Последняя кривая, которую нам надо настроить, голубая, и делается это немного по-другому. Цветопередача в изображении ЛОМО смещена в сторону голубого, но, и это важно, она не однородна по всей кривой. Вместо этого вам нужно создать кривую в форме перевернутого «S», которая поднимается над линией внизу и опускается чуть ниже ее в верхней части. Таким образом некоторые оттенки лишатся голубизны, зато другие ее приобретут. При желании вы можете сохранить свои настройки кривой, чтобы облегчить в дальнейшем применение этого эффекта. Значок + рядом с выпадающим меню Preset [Преднастройка] добавит текущие настройки (для всех цветовых каналов) в список предварительных настроек. Однако на практике, возможно, для достижения желаемого результата будет проще вручную настроить каждое изображение.



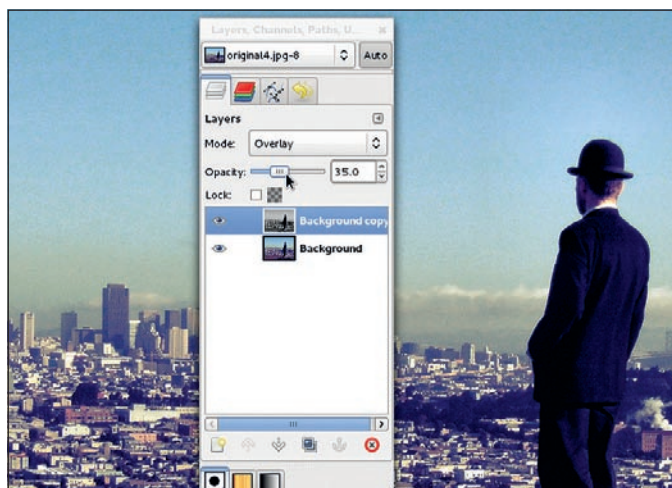
6 Слой Duplicate

Итак, мы получили основу для нашего изображения, но можем слегка добавить общих эффектов и приблизить его к вождленному ЛОМО-изображению, настроив общую яркость. Откройте диалоговое окно Layers из меню или с помощью комбинации Ctrl+L. Теперь щелкните правой кнопкой по активному (и только!) слою, и выберите Duplicate во всплывающем меню. Создается точная копия текущего слоя, которая наложится поверх оригинала. Этот новый слой будет выбираться автоматически, а поскольку это дубликат, вы не увидите на экране никакого отличия.



7 Карта градиентов

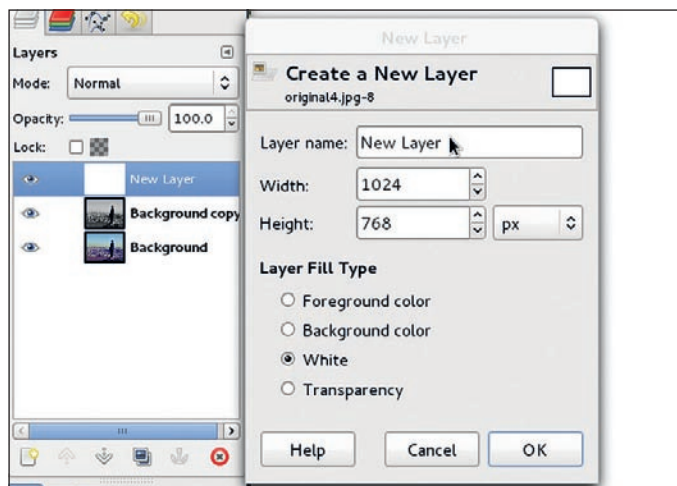
Убедитесь, что у вас выделен верхний слой, и выберите Colours > Map > Gradient [Цвета > Карта > Градиент] из главного меню изображения. Оно использует текущий (черно-белый) градиент для перекраски текущего слоя, и это – быстрый способ конвертировать слой в изображение в серой гамме [Greyscale] (так же, как и с другими эффектами). Конечно, это сработает только в том случае, если градиентом по умолчанию у вас стоит стандартный Black and White [Черно-белый] – вы также можете выбрать Colours > Hue/Saturation [Цвета > Оттенок/Насыщенность] и установить уровень насыщенности на 0. Не меняйте режим Image mode на Greyscale, не то у вас изменятся и другие слои тоже! В выделенном слое перейдите в диалоговое окно Layer [Слой] и выберите Overlay [Перекрытие] в выборе режимов в верхней части диалогового окна.



8 Настройка непрозрачности

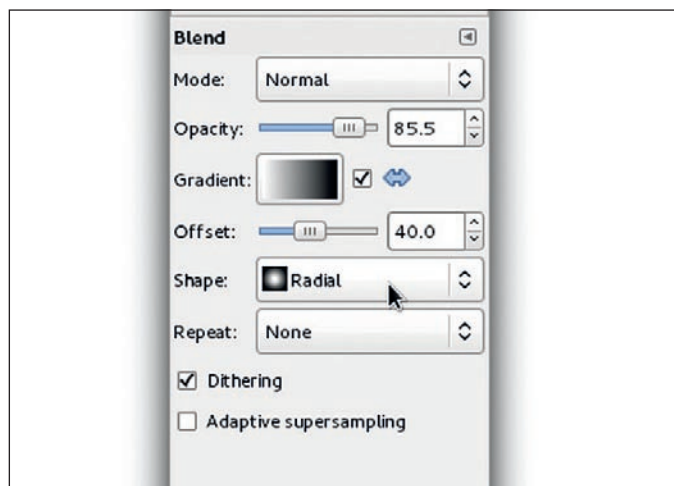
Режим Overlay эффективно затемняет изображение снизу, но не так сильно, как режим Darken [Затемнить], и с ним проще добиться большей точности. Используйте слайдер Opacity [Непрозрачность] для выбора такого состояния, которое будет наилучшим для изображения. Чем сильнее вы его затемняете, тем больше теряется деталей, особенно в плотных областях, но для некоторых изображений в этом и будет состоять желаемый эффект. Обычно нормальным является показатель 40–50%. Конечно, поскольку мы оставляем его в виде слоя, вы всегда сможете вернуться и заново настроить его, если решите несколько шагов, что изображение стало слишком темным.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.



9 Слой виньетирования

Цвета почти настроены, но одним из наиболее примечательных свойств снимков ЛОМО является затемнение по направлению к краям. Причиной тому служит слабая светопередача по краям линз, известная под названием «виньетирование». Можете добиться этого эффекта сами, достаточно просто. Сначала создайте новый слой White поверх уже имеющихся двух. Из главного меню над изображением выберите Layer > New Layer [Слой > Новый слой] и выберите Layer Fill Type > White [Тип заполнения слоя > Белый]. Теперь изображение должно полностью побелеть – но не волнуйтесь, через пару минут мы вернем картинку.



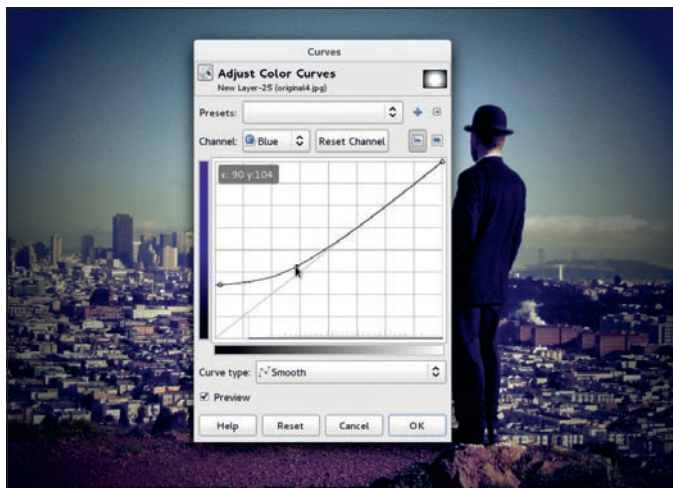
10 Радиальный градиент

В выделенном новом слое переключитесь в режим Multiply. Поскольку слой заполнен белым, это не изменит изображение, которое должно явно проступить. Но мы изменим все это, применив в центре изображения Circular gradient [Круговой градиент]. Нам нужно, чтобы середина была белой, а края – черными, так что выберите инструмент Gradient и измените опции инструмента следующим образом: установите непрозрачность [opacity] на 85%. Щелкните по кнопке рядом с градиентом по умолчанию Black/white, чтобы перевернуть его, как показано. Также выберите режим Radial [Радиальный], который сделает градиент круговым, а не линейным. Предполагается, что вы используете градиент по умолчанию Black/White – если вы изменили свои настройки для GIMP, сначала выберите нужный градиент.



11 Растягивайте из центра

Из приблизительного центра изображения вы можете растянуть градиент в направлении углов. Отпустите его, и проявится виньетирование. Черная часть градиента, располагающаяся в углах изображения, блокирует часть изображения внизу и делает его темнее. Это в точности соответствует тому, что происходит в старых или дешевых камерах, не имеющих хорошей системы линз. Не переживайте, если в первый раз у вас не получится все как надо – вы можете сделать новый градиент сверху, так что пробуйте, пока не начнет получаться. Не добивайтесь эффекта иллюминатора – небольшого затемнения по краям будет вполне достаточно для достижения нужного эффекта.

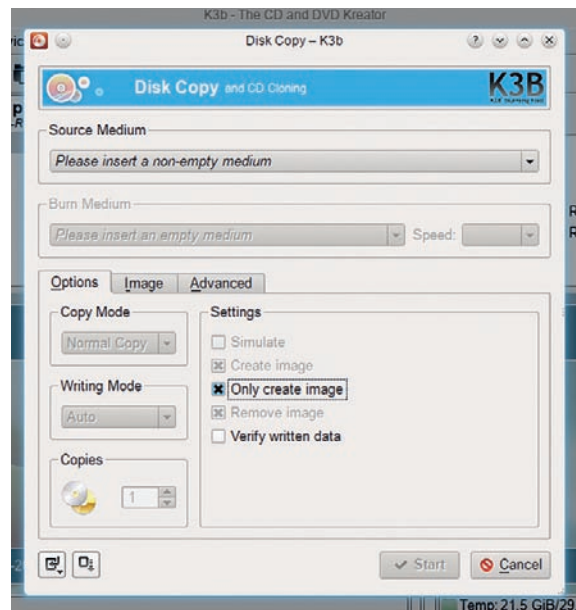


12 Раскрасьте слой

Все теперь выглядит очень неплохо, но черно-белого виньетирования недостаточно для соответствия снимкам ЛОМО, которые имеют тенденцию придавать всему голубоватый оттенок. Это можно легко поправить, если перейти в Colours > Curves и выбрать канал Blue. Перетащите нижнюю часть кривой примерно на треть, но следите, чтобы центр кривой оставался в том же положении, иначе поголубеет все изображение. Работа с параметрами Blue изменит градиент Radial так, что он станет переходить из белого в голубой, а не в черный, и впоследствии этот эффект усилится изображением внизу. Теперь изображение готово, за исключением тех эффектов, которые вы, возможно, захотите добавить – помните, что вы всегда можете вернуться и исправить нужные слои, чтобы добиться желаемого эффекта. Настраивайте все по своему вкусу – и дело сделано! LXF

Конвертируем

Надоев всем соседям своим домашним видео, Джонатан Робертс думает, как показать его всему миру



» K3b позволяет создать ISO-образ любого диска — это отличный вид архивной копии.

Архивация DVD

Для начала разберемся, как заархивировать вашу коллекцию DVD. За годы многие накопили внушительную подборку фильмов, и перенос их с оптических дисков на жесткий — первый шаг к просмотру любимого кино на любом из новейших устройств, будь то iPad, мобильник или еще что. Есть разные точки зрения на степень легальности процедуры конвертирования DVD для личных нужд, и мы здесь не будем обсуждать взлом защиты от копирования. Но наши советы пригодятся тем, у кого есть DVD с домашним или общедоступным видео, например, *Big Buck Bunny* (который мы и возьмем в качестве примера).

Если вы хотите создать идеальную резервную копию DVD для восстановления в случае сбоя, то самое простое — снять ISO-образ DVD-диска.

Это тот же ISO, в котором вы скачиваете новые дистрибутивы Linux, точная копия диска, которую в любое время можно записать на чистую «болванку». В Gnome для этого достаточно вставить диск в привод, щелкнуть по нему правой кнопкой и выбрать Скопировать диск [Copy Disc]. Вместо выбора привода, куда следует производить копирование, укажите Файл образа [File Image], и дело в шляпе. В KDE то же самое выполняется в K3b: на экране Параметры копирования [Copy Disc Settings] есть опция Создать образ [Create Image Only].

Образ диска можно проиграть в медиа-плеерах вроде VLC или в медиа-центрах типа XBMC. Конечно, такие файлы займут немало

Каждый, кто хоть раз видел компьютер, скажет: это штука сложная. Несовместимости, возникающие тут и там, лишь добавляют проблем. Одна из них, изрядно досаждающая пользователям Linux — это мультимедиа-файлы.

По юридическим соображениям, многие дистрибутивы по умолчанию не включают поддержку MP3 — вам нужно выйти в Интернет и заняться этим самостоятельно. А формат, с которым Linux работает «из коробки» — Vorbis — не слишком популярен на портативных плеерах.

Увы, это лишь один пример несовместимости, касающейся мультимедиа. Перейдем в Сеть: из четырех основных браузеров, три поддерживают формат WebM, предложенный Google, а один — H.264. Даже если обратиться к телевизорам и DVD, выяснится, что есть два конкурирующих стандарта, PAL и NTSC, отличающихся по частоте развертки и размеру кадра.

Если вы хотите, чтобы ваши мультимедиа-данные были доступны на различных носителях и платформах, рано или поздно вам придется заняться превращением одного формата в другой. Тут-то вам и пригодится эта статья. Мы проведем вас через джунгли мультимедиа-форматов и познакомим с мощными инструментами-конверторами, чтобы вы справились с преобразованием игрючи.

Мы изучим проблему на конкретных примерах и применим всевозможные утилиты, чтобы вы оценили диапазон их возможностей. В каждом конкретном случае мы не только покажем вам, как работает программа, но и представим наиболее важные настройки, влияющие на размер и качество результата.

«Для начала разберемся, как заархивировать вашу коллекцию DVD.»



Наш эксперт

Супер-скрытый Джонатан Робертс копается в Linux с детских лет. Иными словами, не так уж долго.

прохода. При заданном битрейте, это повышает качество: сначала файл анализируется и выявляются кадры, требующие меньше данных, что позволяет выделить место в общем потоке для других, «подвижных» кадров, требующих больше данных. Единственный минус – кодирование длится вдвое больше времени. В наших тестах с кодированием в два прохода мы не смогли заметить отличий между выходным файлом, сжатым с битрейтом 1400 кбит/сек, и исходным DVD. Даже на уровне 800 кбит/сек заметить отличия будет очень непросто.

Секреты жанра

Еще один момент, который надо иметь в виду при выборе качества кодирования – тип видео, которое вы сжимаете. Если оно содержит динамичные сцены со множеством людей, для получения разумного качества не обойтись без высокого битрейта. Если

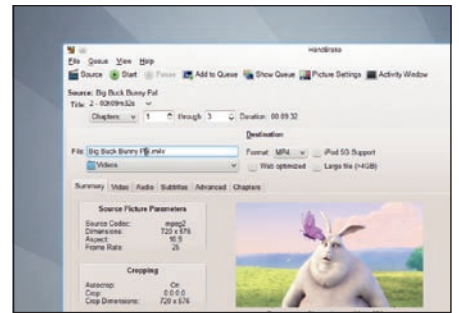
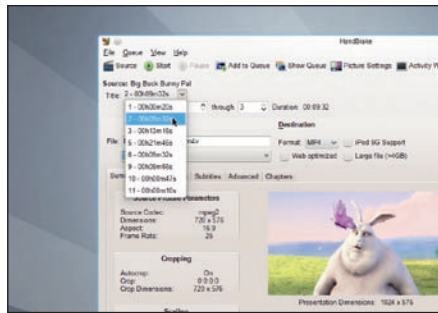
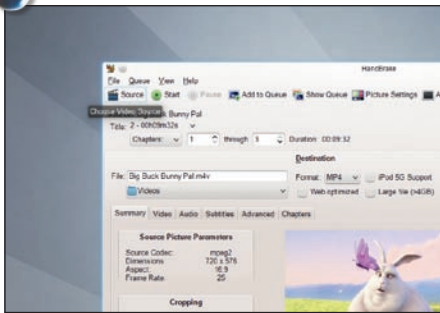
это запись интервью, вас, вероятно, устроит и более низкий. Качество звука настраивается на вкладке **Аудио [Audio]**. Как уже говорилось, выбранный по умолчанию AAC предлагает неплохое качество при сравнительно небольшом размере файла. Но если вы хотите смотреть фильм на устройстве с приличной аудиосистемой – например, DVD-плеере с подключением Hi-Fi – тогда настройки можно подкрутить. 160 кбит/сек, выбор по умолчанию, аналогичен MP3 среднего качества. Высококачественная музыка, продаваемая на Amazon и iTunes, имеет битрейт 256 кбит/сек. По нашему мнению, выше этой точки действует закон убывающего выигрыша, и хотя на хорошей аудиосистеме вы сможете уловить различия до значений порядка 320 кбит/сек, увеличивать битрейт дальше смысла уже нет.

Для справки, при записи звука на DVD редко используется битрейт выше 448 кбит/сек, так что сверх этого замахиваться неза-

«Выше этой точки действует закон убывающего выигрыша.»



Шаг за шагом: Кодлируем видео



1 Укажите привод

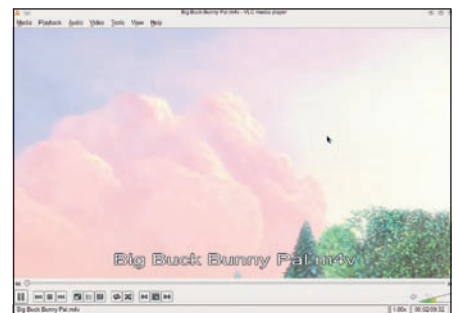
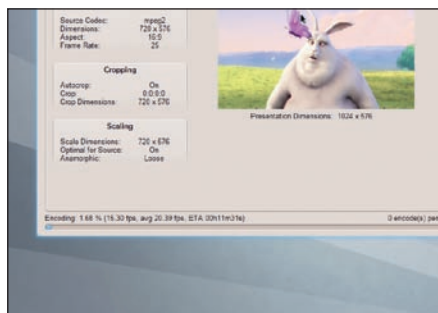
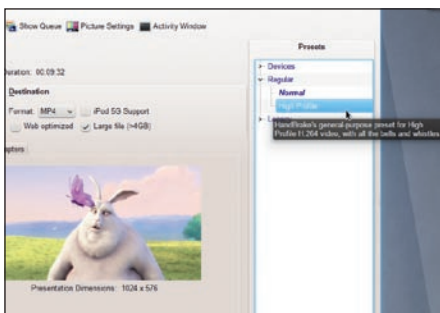
Нажмите кнопку **Источник [Source]** вверху слева и выберите привод в списке **Обнаруженные DVD-устройства [Detected DVD Devices]** внизу окна.

2 Выберите, что копировать

После того, как *Handbrake* изучит диск, вам предложат выбрать ролик. Самый длинный из них и есть главная дорожка. Вам, вероятно, нужна она.

3 Скажите, куда копировать

Далее, задайте местоположение и имя для результирующего файла. Укажите также формат (можно MP4) и активируйте оптимизацию для Web, если ролик будет размещен онлайн, чтобы ускорить потоковое вещание.



4 Определитесь с качеством

Если эта копия делается не для конкретного устройства, выберите что-то из **Стандартных [Regular]** предустановок. Прилагательные **Нормальный [Normal]** и **Высокий [High]** касаются затрат на декодирование: High даст качественный файл малого размера ценой большего потребления ресурсов.

5 Немного расслабьтесь

Настроив все по своему вкусу, нажмите **Пуск [Start]** и приготовьте кофе – или коктейль, как предлагает *Handbrake*. Кодирование требует времени.

6 Дивитесь результату!

Закончив кодирование, взгляните на результирующий файл и удивитесь его размеру. Просмотрите его в *VLC* или другом плеере; и удивитесь качеству!

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

чем. На самом деле, если вы хотите сохранить качество исходного аудио, можно просто выбрать опцию Auto Passthru, извлекающую звук из источника без изменений.

Бывает, что оригинал – не DVD, а, наоборот, файл на жестком диске или карте памяти видеокамеры. Если вы хотите записать его на DVD, чтобы показать отснятое семье и друзьям на большом ТВ, придется конвертировать его в нужный формат. Благодаря утилите *ffmpeg*, это не сложно. Во-первых, убедитесь, что *ffmpeg* есть в системе – не все дистрибутивы включают его по умолчанию; например, в Fedora потребуется активировать репозиторий RPMFusion. Поиск в Интернете подскажет, что следует подключить в вашем случае. После этого – приступайте к кодированию. *Fmpeg* предлагает замечательную опцию, **-target**: с ее помощью можно приказать *ffmpeg* создавать выходной файл, удовлетворяющий требованиям стандартных спецификаций: DVD, VCD и т.п. Чтобы воспользоваться ею, наберите

```
ffmpeg -i /путь/к/исходнику.avi -target pal-dvd /путь/к/результату.avi
```

Вот и все – об остальном позаботится *ffmpeg*! Детали целевого формата можно изменять – главное, чтобы они не конфликтовали со спецификациями. Скажем, если вам нужен битрейт побольше, используйте команду

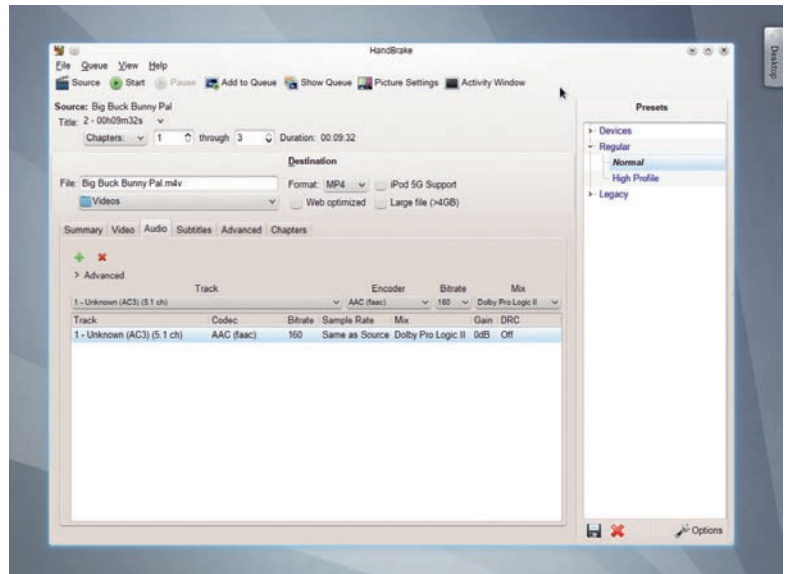
```
ffmpeg -i /путь/к/исходнику.avi -target -pal-dvd -b 2000K /путь/к/результату.avi
```

Так вы создадите файл, совместимый с PAL DVD, имеющий битрейт 2000 кбит/сек. Все, что мы ранее говорили о качестве, остается справедливым (учтите, что PAL – телевизионный стандарт Европы и Азии, а в Северной Америке применяется NTSC).

Видео для Сети

Еще одним, и пожалуй, самым популярным местом для обмена домашними видеозаписями является Интернет. К сожалению, те несовместимости, о которых говорилось в начале статьи, встают здесь в полный рост.

Простейший путь поделиться контентом онлайн – сервисы вроде YouTube или Vimeo. Оба принимают видео различных форматов, так что, кроме случаев, когда вы храните его в ISO или VOB, проблем с выгрузкой быть не должно. Однако и тот, и другой сервисы имеют набор рекомендаций для достижения наилучших результатов. Vimeo предпочитает кодек H.264, битрейт 2000 кбит/сек и размер 640×480 для стандартного разрешения или 5000 кбит/сек и 1280×720 – для высокого (в случае Full HD – 1920×1080).



➤ Вкладка **Аудио** в *Handbrake* позволяет задать кодек и качество сжатия звука. Полиглоты могут даже выбрать дорожку на другом языке.

Как создать видео, удовлетворяющее этим требованиям? Естественно, с помощью *ffmpeg*. Взгляните на команду:

```
ffmpeg -i /путь/к/исходнику.avi -vcodec libx264 -b 2000K -s 640x480 /путь/к/результату.m4v
```

Она создает видео стандартного разрешения, закодированное в H.264 кодеком x.264, с битрейтом 2000 кбит/сек и размером кадра 640×480, в указании которого, правда, нет смысла, если только вы не уменьшаете кадр – в противном случае, повышения качества ждать не приходится.

ffmpeg позволяет даже сконвертировать звук в соответствии с требованиями Vimeo и YouTube:

```
ffmpeg -i /путь/к/исходнику.avi -vcodec libx264 -b 2000K -acodec aac -ab 256K /путь/к/результату.m4v
```

Как нетрудно догадаться, здесь создается выходной файл со звуковой дорожкой в AAC с битрейтом 256 кбит/сек.

Fmpeg поддерживает огромное число форматов. Чтобы узнать, насколько огромное, или уточнить, возможна ли требуемая конвертация, просто выполните **ffmpeg -codecs** и посмотрите, что получится. **LXF**

Видео для HTML 5

YouTube и Vimeo – превосходный способ публиковать видеоролики, но если вы хотите разместить отснятое на собственном сайте и поддерживаете открытые стандарты, вам нужен HTML-тэг **<video>**. Сейчас его «понимают» все популярные браузеры, включая мобильные, и для его работы не требуются подключаемые модули.

Проблема только в том, что каждый браузер поддерживает для **<video>** свою комбинацию кодеков, слегка отличающуюся от остальных. Например, *Safari* и iPhone «понимают» только MP4, с видео в H.264 и звуком в AAC. Но, поскольку H.264 запатентован, *Mozilla*, *Google* и *Opera* не поддерживают его, предпочитая другой стандарт – *WebM*. В нем видео сжимается кодеком VP8, аналогичным по производительности H.264, а звук – кодеком Vorbis. *Internet Explorer* сидит посередине, поддерживая и то, и другое. Итак, вам придется создавать два видео: для *Safari*/iPhone и для всех остальных.

В этом вам поможет последняя версия *ffmpeg* (см. примеры выше). Некоторые дистрибутивы еще собирают *ffmpeg* без поддержки *VP8* и *WebM*, которую можно добавить самостоятельно; см. <http://www.webmproject.org/tools>.

Создав два видеофайла, разместите их на странице, используя следующий HTML-код:

```
<video width='640' height='480' controls>
  <source src='src.webm' type='video/webm; codecs="vp8, vorbis"'>
  ...
</video>
```

Потребуется создать по одной записи **<source>** для каждого формата, в котором вы храните видео.

Обратите внимание также на атрибут **type**; он сообщает браузеру формат видео и необходимые кодеки. Если браузер понимает, что не сможет воспроизвести файл, он, вероятно, не будет его даже скачивать. Допустимые значения параметра

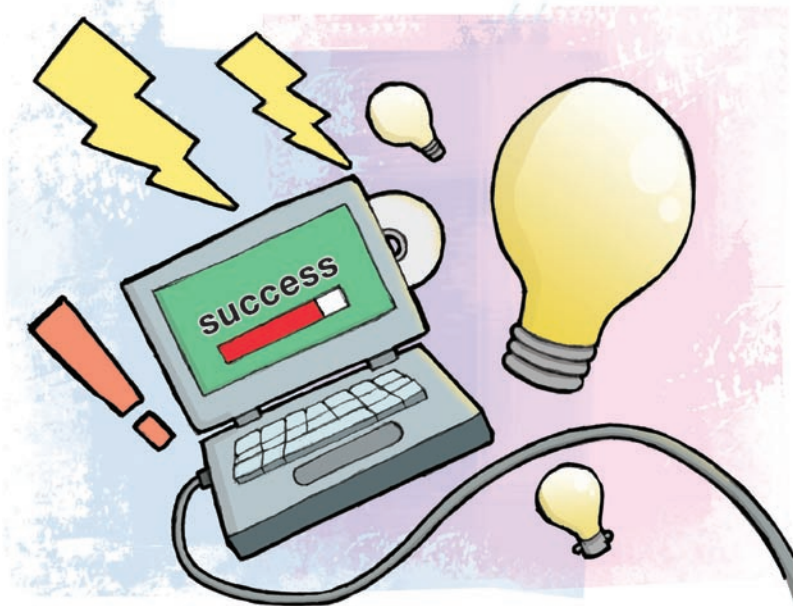
type можно найти на http://wiki.whatwg.org/wiki/Video_type_parameters.



➤ Если вы хотите поразить мир своим роликом, нужен HTML-тэг **<video>**.

Автоматический

Как обеспечить автоматический вход в систему при загрузке компьютера, не ослабив его безопасность? **Нейл Ботвик** знает ответ.



Наш эксперт

Нейл Ботвик
У Нейла стоит по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не выдаст вам местоположение своего центрального сервера.

Большинство дистрибутивов и сред рабочего стола предоставляют средства для автоматического входа в систему от имени указанного пользователя при загрузке компьютера. Это достаточно безопасно, если компьютер находится в безопасном месте – например, у вас дома; но не так уж и полезно, если к нему может подойти кто угодно. Однако если вы хотите настроить компьютер для общего пользования, автоматический вход в систему – именно то, что вам нужно. Как его предоставить и при этом не дать случайному прохожему с чуть более чем обычным знанием Linux шанса злоумышленного использования системы?

Один из вариантов – очищать и перезагружать домашнюю директорию при перезагрузке, перезагружая при необходимости компьютер после каждого пользователя; именно так предложил нам решить эту проблему один из участников нашего форума. Вот что вам надо сделать: создать две учетные записи пользователя, настроить первую по своему усмотрению, а вторую отдать пользователю с автоматическим входом в систему. Затем запустите из `/etc/rc.local` (или его эквивалента в вашем дистрибутиве) скрипт

```
#!/bin/sh
rsync -a --delete /home/default/ /home/user/
chown -R user: /home/user
```

Если ваш компьютер использует **sudo**, не создавайте своего первого пользователя как **guest**: тогда он получит права **sudo**, которые отнюдь не следует давать случайным людям. При таком подходе сразу возникает ряд вопросов по обеспечению безопасности системы. Вам не нужно, чтобы пользователь мог выйти из системы: это предоставит ему попытку войти снова от имени другого пользователя. Вы также должны позаботиться, чтобы выйти из рабочего стола нельзя было ни по **Ctrl+Alt+F1**, ни по **Ctrl+Alt+Backspace**. И еще надо ограничить для этого пользователя опции меню – в частности, никаких ему административных функций.

Однако есть еще один, знакомый вам подход: использовать для системы среду **live CD**. Она будет делать то, что вам нужно, по умолчанию, и автоматически пускать пользователя на рабочий стол со стандартным набором настроек по умолчанию. Вам нужно только подправить эти настройки (начать хорошо бы с удаления опции установки системы на жесткий диск) и перекомпилировать ваш **CD**.

В этом пункте вы можете беспокоиться по поводу использования **CD**. Если компьютер можно загрузить с **CD**, что мешает злодею подменить **CD** и загружаться с чего-то понахальнее? Оптические носители также ограничивают производительность. Если вам случалось использовать **live CD**, вы знаете, насколько он медленнее системы на жестком диске, в частности, при загрузке.

С обеими проблемами справиться легко. На **Live CD** находятся всего три файла: ядро, начальный **ramdisk** и образ корневой файловой системы **root filesystem** (и, конечно, **bootloader**, чтобы все это загрузить). Нигде не сказано, что эти файлы обязаны жить на **CD** или **DVD**: они с тем же успехом могут сидеть и на жестком диске. Это сделает систему безопасной, поскольку **root filesystem** предназначена только для чтения [**read-only**], почти такой же быстрой, как стандартная установка на жесткий диск, и при перезагрузке все настройки будут сбрасываться в значения по умолчанию.

Как же создать такую систему? Первый шаг – установить ваш любимый дистрибутив на жесткий диск. Не обязательно делать это на том компьютере, где будет работать окончательная система. Мы для этого упражнения использовали **Mint 12**, отчасти потому, что это – отличный дистрибутив, а отчасти потому, что нас попросили об этом на форуме.

Первый шаг – установить **Remastersys**, сначала добавив его источник в репозитории командой в терминале

```
echo "deb http://www.geekconnection.org/remastersys/repository lucid/" | sudo tee -a /etc/apt/sources.list.d/remastersys.list
```



» Если вам нужно готовое решение, обратите внимание на прекомпилированные дистрибутивы для интернет-киосков, наподобие **Webconverger**.

ВХОД



Затем вы сможете установить программу через *Synaptic*. Вы увидите предупреждение, что программа не может быть аутентифицирована. Это нормально, потому что вы не добавили в репозиторий ключ GPG. Между прочим, мы использовали *Remastersys* для создания версии Ubuntu 11.10 с Gnome 3 на **LXF DVD155**. Если у вас не Debian-подобный дистрибутив, пакеты и исходник имеют на <http://www.geekconnection.org/remastersys>.

Следующий шаг – должная настройка системы. Создайте отдельного пользователя для автоматического входа в систему:

```
sudo useradd -m kiosk
sudo passwd kiosk
```

Войдите в систему от имени этого пользователя и настройте рабочий стол. Эти настройки будут применены в финальном образе. Теперь отредактируйте `/etc/lightdm/lightdm.conf`, установите `autologin-guest` в `true` и прокомментируйте запись `auto-login-user`. Эти действия вызовут загрузку для пользователя `guest` прямо на рабочий стол, с установками, унаследованными от `/etc/skel`, и чистой домашней директорией. Нам нужно подправить настройки `bootloader`, чтобы никто не смог перехитрить вашу систему безопасности, например, загрузившись в режиме `Single User`. Отредактируйте `/etc/grub.d/40_custom`, добавив две строки:

```
set superusers="fred"
password fred barney
```

Это добавит пользователя и даст ему пароль. В отсутствие других директив это означает, что если вы попытаетесь отредактировать запись в меню или использовать оболочку *Grub*, вам понадобится это имя и пароль. Вы также можете защитить паролем отдельные записи в меню в *Grub*, приписав

```
--users username
```

в их разделы `menuentry`. Возможно, вы также захотите отредактировать `/etc/default/grub`, уменьшив величины `GRUB_HIDDEN_TIMEOUT` и `GRUB_TIMEOUT` – пары секунд вполне хватит. Затем запустите `update-grub`, чтобы применить новые установки.

Запуск Remastersys

Теперь запустите *Remastersys* через меню Applications > Other или `remastersys-gtk` из терминала. Вы можете настроить некоторые опции во вкладке Settings; обычно настройки по умолчанию вполне нормальные, но можно отключить опцию размещения значка установки на рабочем столе или включить список разделенных пробелом файлов, которые вы хотите убрать из финального образа. Возможно, вы решите исключить такие вещи, как *Synaptic*, или какие-то программы терминала.

Во вкладке Action выберите пользователя, которого вы настраиваете, `kiosk`, как описано выше, чтобы применились его настройки, затем щелкните по `Distcdifs`, который создает файлы `live CD`, но не пишет их в файл `ISO`. Желая проверить запуск в виртуальной машине, тоже используйте опцию `Dist` для создания `ISO`.

Теперь у вас есть рабочий `live CD`, автоматически перенастраивающий все при перезагрузке, не давая пользователю навредить системе; заставим его загружаться с жесткого диска. Лучше использовать для этого отдельный диск – большой объем тут не требуется. Настройте раздел, и, возможно, небольшой раздел подкачки `swap`, и установите на диск *Grub* командами

```
mkdir -p /mnt/kiosk/boot/grub
grub-install --boot-directory=/mnt/kiosk/boot
```



➤ Ваш пользователь `guest` может видеть менеджер программ, но ничего не может с ним сделать, не имея прав `sudo`.

предполагая, что вы подмонтировали раздел `root` на `/mnt/kiosk`. *Remastersys* сохранил файлы `CD` в `/home/remastersys/remastersys/ISOTMP`. Скопируйте или переместите директорию `casper` отсюда в `/mnt/kiosk` и добавьте в `/etc/grub.d/40_custom`, после строк пароля

```
menuentry "Kiosk" {
set root='(hd0,msdos1)'
linux /casper/vmlinuz boot=casper quiet splash --
initrd /casper/initrd.gz
}
```

Настройки `root` здесь для первого раздела на первом диске – укажите их в соответствии со своей раскладкой. Здесь взяты те же опции, что и в файле `isolinux.cfg` для `CD`. Добавьте это в `/etc/default/grub` – новая запись станет параметром по умолчанию.

```
GRUB_DEFAULT="Kiosk"
grub-mkconfig -o /mnt/kiosk/boot/grub/grub.cfg
```

пишет файл меню *Grub* в нужное место на жестком диске (`update-grub` – это просто скрипт, который запускает `grub-mkconfig`). Теперь удалите в `/mnt/boot/grub/grub.cfg` раздел `10_linux` и строку, начинающуюся с

```
search --no-floppy --fs-uuid --set=root
```

Перезагрузитесь с этого диска, чтобы запустить свою защищенную от вторжения установку для киоска. Зайдите также в настройки BIOS, чтобы отключить загрузку с USB или оптических дисков и задать пароль для дальнейшей правки настроек BIOS. **LXF**

Шаг за шагом

- 1 Добавим репозиторий *Remastersys*** нет в стандартных репозиториях `Ubuntu` или `Debian`, вам нужно добавить его запись в список источников.
- 2 Установим *Remastersys*** Добавив репозиторий, установите *Remastersys* через *Synaptic* или `apt-get`, как любую другую программу.
- 3 Выберем пользователя** Выбор пользователя будет означать, что по умолчанию будет использована домашняя директория этого пользователя, позволяя вам индивидуально настроить начальный рабочий стол.
- 4 Настроим ряд опций** Отключите опцию отображения значка установки на рабочем столе в *Remastersys*. Также можете перечислить те программы, которые вы не хотите включать.
- 5 Создадим файлы** Опция `Distcdifs` создает файлы, которые идут на `CD` без создания `ISO`-образа. Используйте `Dist`, если хотите протестировать загрузку с `CD`.
- 6 Вот и все!** В результате загружается готовая система. Настройки рабочего стола соответствуют пользователю киоска, даже если вы войдете в систему как `guest`.

Arduino: СВЯЗЬ

Соедините пару плат Arduino в два процессора или сделайте удаленный датчик с радиоканалом. Большие связи **Ника Вейча** все это позволяют.

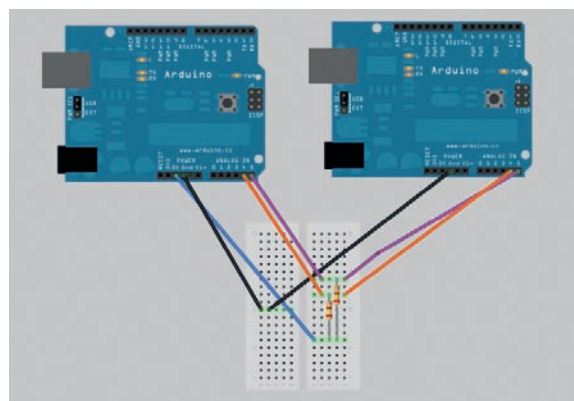


Рис. 1а. Соединить Arduino через шину I²C просто — иногда даже не нужны повышающие резисторы.

```
Wire.send(0x01);  
Wire.endTransmission();
```

Первая строка инициализирует это устройство как главное. Затем ему нужно поговорить с определенным устройством, и мы загружаем его адрес. Данные обычно отправляются байтами, но в качестве параметров метода **send()** можно использовать указатель и длину. Наконец, всегда стоит приятно завершать разговор — **endTransmission()** на самом деле иницирует передачу данных, которые были помещены в буфер методом **send()**. Чтобы получить данные с ведомого устройства, главное иницирует разговор и говорит, сколько байтов оно хочет получить:

```
char buffer[8];  
Wire.requestFrom(slave_address, 8);  
int count=0;  
while(Wire.available())  
{  
    buffer[count] = Wire.receive();  
    count++;  
}
```

Функция **Wire.available()** возвращает количество принятых байт, находящихся в буфере в данный момент — для их получения используется метод **Wire.receive()**. Можно и вернуть в эту передачу добавочный код, чтобы убедиться в получении нужного количества данных или получить фрагменты, потерявшиеся по дороге.

В порядке более полезного примера напишем короткую программу «пинга». Она установит одно устройство как главное, другое — как подчиненное. Можно использовать один и тот же код для обоих устройств и задать главное устройство на аппаратном уровне, просто подключив один из цифровых выходов к +5В и проверить его внутри программы.

Вы не всегда предпочтете писать код таким образом, но для проверки гораздо проще иметь один блок кода — оказывается, ведомому устройству многое все равно не нужно. «Железа» тоже много не потребует. На рис. 1а показано, как это делается физи-



Наш эксперт

Когда **LXF** только появился, его держали на плаву исключительно скрипты *Bash* от **Ника Вейча**. Потом их заменили «люди», и это, по мнению Ника, стало шагом назад...

Есть куча причин, по которым вашему проекту может понадобиться больше одного Arduino — возможно, у вас есть массив удаленных датчиков или вы хотите управлять роботом на базе Arduino с другого Arduino. В любом случае, без взаимодействия тут не обойтись. К счастью, организовать его довольно просто. Способов передачи и приема данных много, и выбор зависит от ваших потребностей — в частности, от расстояния.

Экономим провода

Интерфейс SPI быстр и хорошо поддерживается в Arduino, но требует жуткого количества проводов — на него тратится по меньшей мере четыре вывода (или больше, для адресации). Тут еще есть смысл при беспроводном соединении, но если платы связаны локально, более чем достаточно двухпроводного интерфейса I2C.

Библиотека *Wire* (см. наши более ранние эксперименты с EEPROM в **LXF152/153**) позволяет просто связать два Arduino, сэкономя множество дополнительных выводов. В ней надо задействовать аналоговые выводы 4 и 5, но это тоже просто.

Используется конфигурация «ведущий/ведомый», шиной управляет одно устройство, и никто не заговорит, пока к нему не обратятся первым. Теоретически на одной шине может быть множество плат Arduino или других устройств, управляет которыми одно главное. На практике, без применения специальных аппаратных повторителей длина создаваемой шины ограничена. Чтобы поговорить с устройством, выполните следующие шаги:

```
Wire.begin();  
Wire.beginTransmission(Slave_address);
```

» **Месяц назад** Мы преспокойно изготовили Arduino сами.



ДВУХ ПЛАТ

чески (если у вас хорошие проводники, на макете можно обойтись без повышающих резисторов и подключить платы друг к другу напрямую). **Рис. 1b** проясняет эту схему, и шину можно расширить для подключения других устройств. Разобьем наш тестовый код на фрагменты (полностью он приведен на DVD в файле `i2c_ping`):

```
#include <Wire.h>
const int configpin = 7;
bool config;
unsigned char buffer[8];
int Slave=8;
```

В начале программы импортируется нужная библиотека и объявляются переменные – несколько для описания нашей схемы аппаратной конфигурации, затем буфер данных и адрес подчиненного устройства, с которым мы будем общаться.

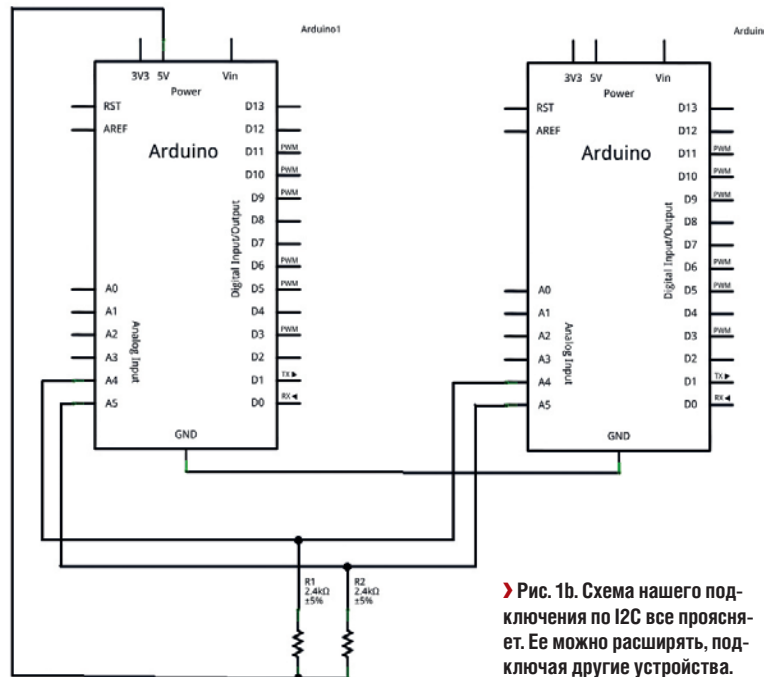
Код настройки по состоянию вывода определит, является ли устройство ведущим или ведомым, и установит все как надо:

```
void setup(void)
{
  pinMode(configpin, INPUT);
  config = digitalRead(configpin);
  Serial.begin(9600);
  Serial.print("Initialised as:");
  if (config){
    Serial.println("transmitter");
    Wire.begin();
  }
  else{
    Serial.println("receiver");
    Wire.begin(Slave);
    Wire.onReceive(slaveRX);
    Wire.onRequest(slaveTX);
  }
}
```

Код в общем понятен. Самое интересное происходит во фрагменте для ведомого устройства. Существует два метода установки `callback`-функции (обратного вызова) – в одном она вызываетея, когда библиотека `Wire` фиксирует отправку данных ведущим устройством, а в другом – когда принимается запрос на возврат данных. Этим мы пока не занимались.

В главном цикле должен быть код только для главного ведущего, но в качестве обратной связи мы можем отправлять какие-то сообщения о состоянии по последовательному каналу при возникновении определенных событий. Основной код здесь передает байт данных, а затем просит передать его обратно. С помощью встроенного таймера (путем вызова функции `millis()`) можно узнать, сколько времени занимают отправка и возврат:

```
void loop(void)
{
  if (config) {
    uint32_t time = millis();
    uint32_t start;
    bool timeout = false;
    Serial.print("Now sending ");
```



» Рис. 1b. Схема нашего подключения по I2C все проясняет. Ее можно расширять, подключая другие устройства.

```
Wire.beginTransmission(Slave);
Wire.send(0xFF);
Wire.endTransmission();
О методах для передачи данных мы говорили выше. Теперь, когда данные отправлены, можно попросить прислать их обратно:
Serial.println("Waiting for response");
delay(20);
Wire.requestFrom(Slave,1);
delay(20);
Serial.println("Data requested");
buffer[0]=Wire.receive();
Serial.print("response received");
Serial.println(buffer[0],HEX);
Serial.print("round-trip:");
Serial.println(millis()-time);
delay(2000);
}
else
{
  Serial.print("waiting");
  delay(1000);
}
}
```

Наконец, нужно написать функции для работы на ведомом устройстве. Каждая из них вызывается с аргументом – целым числом, которое задает число принятых или запрошенных байт:

```
void slaveRX(int bytes){
  buffer[0]=Wire.receive();
```

ЧТО НАМ НАДО

» Две платы Arduino (любых).

» Для схем с радиосвязью вам понадобятся две платы nRF2401+ (см. <http://www.sparkfun.com/products/152>, <http://proto-pic.co.uk/transceiver-nrf2401a-with-chip-antenna/>, или поищите на eBay).

» Библиотека RF с LXF DVD.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

```
Serial.print("value received:");
Serial.println(buffer[0],HEX);
}
void slaveTX(){
Serial.println("sending reply");
Wire.send(buffer[0]);
}
```

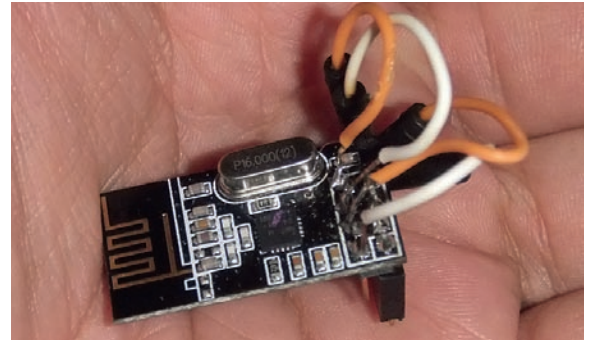
Здесь есть несколько задержек, чтобы линии могли приходить в исходное состояние между передачами данных – на хороших шинах этого не потребуется; может, впрочем, оказаться, что нужно несколько повышающих резисторов, если напряжение на линиях будет с трудом достигать 5 В. Подключите к каждой линии на +5 В резистор сопротивлением 2 кОм. На нашем тестовом оборудовании на отправку и прием байта ушло около 90 мкс без задержек. Не так уж плохо.

Если вам нужен беспроводной канал связи, вариантов несколько. Из дорогих – стандарт «ZigBee», поддерживаемый схемами и модулями XBee, предоставит вам все, о чем вы мечтаете. В большинстве схем XBee применяется простой последовательный интерфейс, и они часто используются в библиотеках и примерах Arduino. Большой недостаток – стоимость: около 20 фунтов за устройство – это не то, что вы охотно приплюсуете к каждому проекту. Из более доступных – платы радиосвязи, работающие по принципу регенерации (для щеголяющих старомодностью, поясню: «автодины»). Они достаточно дешевы, и их легко собрать или переделать самим. Но им не хватает оснастки – придется писать собственные протоколы для отправки и приема данных; вдобавок эти устройства способны интерферировать друг с другом, и пользоваться несколькими устройствами по соседству может быть затруднительно. «Золотая середина» – серия трансиверов RF24XX. Производимые Nordic Semiconductors, эти чудесные маленькие схемы работают на частоте 2,4 ГГц. Немного не дойдя до полноценного сетевого протокола, они предлагают столь полезные возможности, как выбор канала, передача пакетов подтверждения и различные скорости передачи (чтобы выжать из сигнала наибольшее расстояние), и обойдутся вдвое дешевле XBee.

Эти микросхемы можно купить уже смонтированными на оконечной плате [breakout board] вместе с антенной. Такие платы выпускает Sparkfun. Их довольно сложно пристроить к Arduino, но скоммутировать все на макетной плате не составит труда.

Схемы nRF24XX являются полудуплексными. Они могут отправлять или принимать данные, но не одновременно. Чтобы реализовать это программно, придется немного повозиться, но эта проблема свойственна и другим (не по радиоканалу) способам соединения. В любом случае, микросхемы ATmega для многозадачности не совсем пригодны.

Существует пара реализаций библиотеки и для этого устройства. Более сложная из двух, но и с большей функциональностью – библиотека RF24 от Джеймса Колиза-младшего [James Coliz, Jr]. Она поддерживает многие аппаратные функции микросхемы без лишних осложнений. Посмотрим, как с помощью этой



► Не все микросхемы для создания канала удобно подключать к макету.

библиотеки настроить соединение по радиоканалу и воспользоваться им:

```
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
```

Эта библиотека использует библиотеку SPI для Arduino, поэтому ее тоже нужно подключить. Здесь также подключается класс RF24 – с его помощью легко создать радиоканал, и после подключения всего этого мы создаем экземпляр радиоканала:

```
RF24 radio(8,9);
```

Здесь инициализируется объект radio с использованием выводов 8 и 9 Arduino в качестве выводов CE (Chip Enable – микросхема активна) и CSN (Chip Select Not – микросхема не выбрана) соответственно. Название «Микросхема не выбрана» может показаться забавным, но по сути это инвертированный вывод «Микросхема выбрана» (Chip Select) – он активен в состоянии «нуля», и микросхема выбирается, когда этот вывод соединяется с «землей». Это обычное дело для интерфейсов SPI, поэтому даже если вывод отмечен как CS или SS, проверьте, активируется ли он «нулевым» уровнем. Другие выводы для обмена данным по SPI – те, что обычно используются библиотекой SPI в Arduino: 11, 12 и 13.

В микросхемах nRF24 есть набор внутренних каналов для приема и передачи данных. С помощью одного канала можно принимать или отправлять данные, с помощью еще пяти – только слушать. У каждого канала есть адрес – нечто вроде MAC-адреса в сети. Это 64-битное число, и оно должно быть максимально случайным, чтобы избежать конфликтов. Поэтому ваша микросхема может передавать данные на один заданный адрес и принимать данные с пяти адресов. Единственное ограничение – в том, что первые четыре байта адреса для всех принимающих каналов должны быть одинаковыми. Чтобы задать 64-битное число в нашем коде, мы используем 64-битное беззнаковое целое число (добавляем LL (long long) в конец шестнадцатеричного значения числа):

```
const uint64_t txpipe = 0x818181818101LL;
const uint64_t rxpipe1 = 0xFFFFFFFFF01LL;
const uint64_t rxpipe2 = 0xFFFFFFFFF02LL;
```

Скорая помощь

А что у нас с однопроводными соединениями? Некоторые датчики используют именно такое соединение для передачи данных (см. www.arduino.cc/playground/Learning/OneWire), но для обмена большими объемами данных между двумя Arduino это не лучший вариант.

Метод	Преимущества	Недостатки	Реальная дальность
Двухпроводное соединение (I2C)	Просто, быстро, и к шине можно подключить несколько устройств.	С кабелем большой длины может не работать; медленнее, чем SPI в Arduino.	В пределах одного стола.
Трехпроводное соединение (SPI)	Работает на больших расстояниях по сравнению с I2C, легко написать код с помощью библиотеки SPI.	Требуется больше проводников, и для каждого подчиненного устройства нужен отдельный проводник.	В соседней комнате.
Последовательный порт (UART)	Простота, работает на довольно длинных расстояниях.	Медленный, может оказывать влияния на плату перепрограммирования; только одно соединение.	В гостиной.
RS485	Хорошие результаты и высокая скорость по длинному кабелю.	Нужны дополнительные устройства и хороший кабель; также загружает последовательную шину.	В пределах дома.
Беспроводное соединение	Без проводов! Не так дорого, как вы думали – классно!	Более сложная настройка; может быть менее надежным, требуется много проводов.	Внизу в саду и дальше.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

В коде установке мы инициализируем объект radio:

```
radio.begin();
и настраиваем различные каналы:
radio.openWritingPipe(txpipe);
radio.openReadingPipe(1,rxpipe1);
radio.openReadingPipe(2,rxpipe2);
```

Функция **openReadingPipe()** принимает два аргумента. Первый – номер используемого канала. Канал для чтения 0 используется и для записи, поэтому если вам на самом деле не нужно шесть каналов, лучше пропустить его. Хотя мы задали адреса каналов, сам класс radio ничего не делает. Нам нужна другая команда, чтобы заставить его слушать:

```
radio.startListening();
```

При приеме данных классом radio он заполняет буфер самой микросхемы. Чтобы получить данные в нашей программе, нужно периодически проверять, есть ли там данные, и считывать данные в собственный буфер. Очевидно, что это нужно делать в главном цикле:

```
uint32_t data;
if (radio.available()) {
    radio.read( &data, sizeof(data) );
}
radio.stopListening();
```

Здесь мы объявляем 32-битное целое для приема данных (данные передаются в пакетах по 32 бита). Мы проверяем, поступили ли какие-то данные, с помощью метода **available()**. Если да, мы их считываем! Этот метод принимает адрес, по которому нужно сохранить данные, и размер данных, которые нужно считать (в 8-битовых байтах). Можно было сразу записывать по 4 байта, чтобы сэкономить время, но вспомогательная функция **sizeof()** уменьшает шансы ошибиться.

Метод **radio.read()** также возвращает булевское значение, и если данные еще есть в буфере, то это True. С его помощью можно построить цикл для считывания всех доступных данных. Прекратив ожидание данных, мы также вызываем метод **stopListening()**, который по сути дела переводит радио в спящий режим. Из-за полудуплексного обмена данными нужно выключить радио, чтобы записать в него данные.

```
uint8_t output = 128;
bool ok = radio.write( &output, sizeof(output) );
```

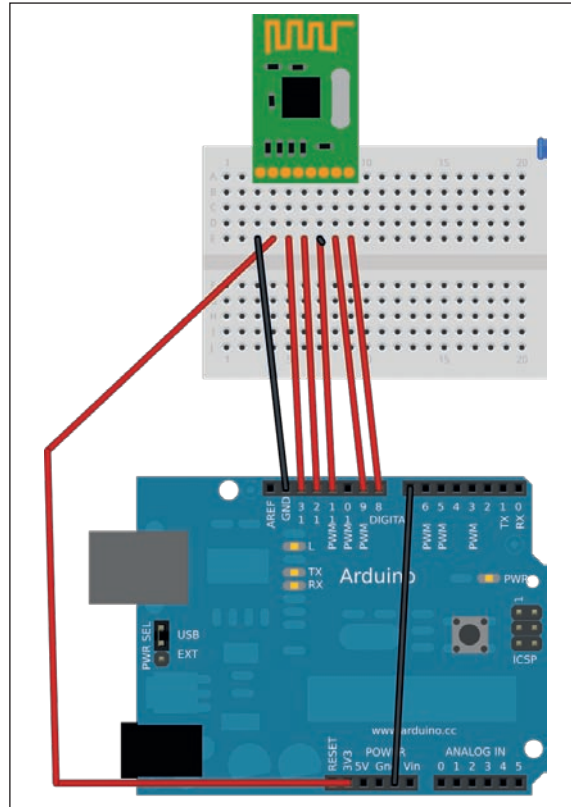
Пакеты ACK

Великие умы, спроектировавшие nRF2401, предусмотрели особый режим подтверждения. Так как радиоприемник после приема должен отправить пакет назад, чтобы подтвердить получение данных (если вы не отключили эту возможность), почему бы не записать что-нибудь в этот пакет? Если определить одиночный пакет для возврата, он будет отправлен обратно и декодирован обычным образом.

Один из вариантов применения этой схемы – удаленный датчик. Вместо того, чтобы писать код для отправки назад значения при каждом запросе на него, можно просто сохранить текущее значение датчика в пакете ACK, и оно будет возвращено автоматически – и это освобождает вас от необходимости выключать режим прослушивания, чтобы отправить ответ. Вы просто записываете данные в пакет таким образом:

```
radio.writeAckPayload( 1, &data, sizeof(data) );
```

Впрочем, в определенный момент нужно очистить буфер FIFO, считав из него принятые пакеты, иначе он переполнится и перестанет выдавать подтверждения для принимаемых данных.



➤ На большинстве плат nRF24xx есть встроенная антенна. Она не дает лучшую дальность – можно также приобрести платы с разъемами для антенн, как у Wi-Fi.

```
if (ok)
    Serial.println("ok...");
else
    Serial.println("failed.");
```

Метод **radio.write()** очень похож на метод приема данных – он принимает адрес данных и размер данных для отправки, в данном случае всего 1. Он также возвращает булевское значение, означающее, была ли принята передача. Независимо от вашего кода, в микросхемах nRF для проверки приема данных отправляются пакеты подтверждения (ACK). Существует возможность автоматической повторной передачи – в этом случае после небольшой задержки будет предпринята повторная попытка отправки каждого пакета. Ее настройки можно задать таким образом:

```
radio.setRetries(15,15);
```

Здесь первое число – задержка в блоках по 250 мкс (максимальное количество блоков – 15; $250 \times 15 = 4000$ мкс), а второе – количество попыток (максимальное – 15).

Если у вас есть трудности с отправкой и приемом данных, не считая проверки подключения, это может быть вызвано интерференцией. В радиопередатчике доступны 127 частотных каналов – некоторые из низших частот могут конфликтовать с Wi-Fi, поэтому выберите канал с номером побольше (на обоих устройствах).

```
radio.setChannel(111);
```

Итак, переписав наш код «пинга» для RF24 (см. DVD), мы получили следующие результаты: производительность осталась хорошей, время прохождения байта составило около 25 мкс со всеми накладными расходами – это в четыре раза быстрее 1°C.

Как бы вы ни решили подключить друг к другу платы Arduino, помните, что сеть не идеальна. Данные иногда теряются, пакеты не доходят, а провода грызут мыши. Вам нужно все предусмотреть, чтобы ваш код не блокировался. Пишите код с умом, и помните, что неожиданности случаются всегда. **LXF**

Скорая помощь

Вам не всегда нужно два Arduino! Похожие возможности могут предоставить и другие (дешевые) члены семейства Atmel – их в любом случае достаточно для передачи данных датчика по последовательному соединению. Чтобы познакомиться с ними поближе и запрограммировать их, вам придется окунуться в мир *gcc-avr* и *AVRdude* (см. **LXF156**).

» **Через месяц** Припасайте беруши! Мы будем создавать музыку (какофонию).



Erlang: Магия

Андрей Ушаков завершает «большое колдовство» с битовыми строками, раскодируя их.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

В этом номере мы продолжим обсуждение большого примера использования битовых строк: рассмотрим, каким образом объекты Erlang, согласно правилам **ASN.1 BER** (то, чем мы занимались в прошлом номере), раскодировать обратно.

Первый шаг, как и во всех других примерах – определение модуля с функциональностью примера, подключение файлов с определениями и определение списка экспортируемых функций. Экспортируем мы всего две функции: **build/1** для построения функции диспетчера, которая выбирает должную функцию для декодирования данных по их типу, и **decode/2**, для самого декодирования.

```
-module(asn1_decoder).
-include("asn1_tag.hrl").
-export([build/1, decode/2]).
```

Функция-диспетчер использует список пар (кортежей из двух значений) из типа данных и соответствующей ему функции для декодирования объекта этого типа. При создании функции-диспетчера мы используем как внутренний (заданный нами) список пар из типа данных и соответствующей ему функции для декодирования, так и внешний (задаваемый пользователем функции через единственный параметр) список таких пар.

Внутренний список пар обрабатывает общие ситуации декодирования (когда объект является списком, кортежем, целым числом и т.д.); внешний список (задаваемый пользователем) служит для обработки специфичных ситуаций декодирования – например, если мы хотим декодировать записи отличным от обычных кортежей образом. Для этого пары из внешнего списка идут всегда перед парами из внутреннего списка, что означает более высокий их приоритет. Более того, если одна пара идет перед другой, то это означает, что приоритет у этой пары выше, т.к. эта пара будет использована в процедуре выбора подходящей функции декодирования первой.

Вполне очевидно, что внешний список пар, используемый для построения функции-диспетчера для декодирования объектов по их типу, должен соответствовать списку пар функций, используемый для построения соответствующей функции диспетчера для кодирования объектов (об этом мы говорили в предыдущем номере). Соответствие этих списков означает следующее: функции, кодирующие и декодирующие одни и те же типы данных должны располагаться одинаково в соответствующих списках. Если не соблюдать соответствие списков, то возможны ситуации, когда невозможно декодировать закодированные объекты, либо когда мы кодируем объект одного типа, а при декодировании получаем объект другого типа, отличного от исходного.

```
build(ExternalDecoders) when is_list(ExternalDecoders) ->
InternalDecoders =
[
{#tag{class = universal, form = primitive, tag_value = 1}, fun
decode_boolean/2},
{#tag{class = universal, form = primitive, tag_value = 2}, fun
decode_integer/2},
{#tag{class = universal, form = primitive, tag_value = 9}, fun
decode_real/2},
{#tag{class = universal, form = primitive, tag_value = 4}, fun
decode_octetstring/2},
```

```
{#tag{class = universal, form = primitive, tag_value = 3}, fun
decode_bitstring/2},
{#tag{class = universal, form = constructed, tag_value = 16},
fun decode_sequence/2},
{#tag{class = universal, form = constructed, tag_value = 32},
fun decode_tuple/2},
{#tag{class = universal, form = primitive, tag_value = 33}, fun
decode_atom/2}
],
TotalDecodersList = ExternalDecoders ++ InternalDecoders,
fun(Binary, DecodeDispatcher) ->
{Tag, _BinaryRest} = decode_tag(Binary),
case lists:keyfind(Tag, 1, TotalDecodersList) of
{Tag, Decoder} -> {ok, Decoder(Binary, DecodeDispatcher)};
false -> false
end
end.
```

Функция **decode/2** использует функцию-диспетчер декодирования, которую мы построили при помощи функции **build/1**; эту функцию-диспетчер мы передаем в качестве первого параметра (в качестве второго параметра передается битовая строка, содержащая закодированный объект). Работа функции **decode/2** основана на использовании функции-диспетчера декодирования: мы вызываем функцию-диспетчер, передавая в качестве параметров декодируемый объект и саму функцию-диспетчер декодирования. Если передаваемые данные могут быть декодированы в объект, то будет возвращен кортеж, состоящий из атома **ok** и декодированного объекта; если же передаваемые данные не могут быть декодированы, то будет возвращен атом **false**. В последнем случае, мы генерируем исключение времени выполнения.

Следует также сказать, что в отличие от функции кодирования объектов **encode/2** (которую мы приводили в предыдущем номере), для работы функции **decode/2** не нужна функция для поиска первой подходящей пары, наподобие функции **first/3** (опять же приведенной в предыдущем номере). Это связано с тем, что в функции **decode/2** поиск пары происходит по ключу (которым является тип данных), и для этой операции достаточно функции **lists:keyfind/3**. В отличие от функции **decode/2**, в функции **encode/2** поиск первой подходящей пары производился в списке пар функций, и критерием окончания поиска было нахождение пары, первая функция которой возвращала атом **true** для исходного объекта. Такая функциональность не реализована ни среди функций модуля **lists** (модуля работы со списками), ни среди функций любых других стандартных модулей.

```
decode(Value, DecodeDispatcher) ->
case DecodeDispatcher(Value, DecodeDispatcher) of
{ok, Result} -> Result;
false -> erlang:error(unsuitable_value)
end.
```

Теперь перейдем непосредственно к декодированию данных. В первую очередь нам необходимо уметь декодировать тип данных (он же тэг данных). Связано это с тем, что по типу данных мы выбираем потом подходящую функцию для декодирования самого объекта. Как уже говорилось, тип данных состоит из трех компонент:

БИТОВЫХ СТРОК 3

класса, формы и идентификатора типа данных. Класс и форма имеют фиксированный размер – 2 и 1 бит соответственно. Способ кодирования идентификатора типа данных зависит от его значения. Если значение идентификатора меньше 31, то он занимает оставшиеся 5 бит октета (байта) полностью. Если же его значение больше или равно 31, то в оставшиеся 5 бит октета записывается значение **2#11111**, после чего идет значение идентификатора, закодированное более сложным способом, о котором мы поговорим ниже. За декодирование типа данных отвечает метод **decode_tag/1**. Входящий параметр у него один – битовая строка с данными для декодирования. Входящий параметр в заголовке метода при помощи операции соответствия шаблону [pattern matching] разбивается на 2 бита для класса, 1 бит для формы, 5 битовый сегмент (либо для идентификатора типа, либо для величины **2#11111**) и оставшуюся часть битовой строки. Значение 5 битового сегмента определяет, можно ли сразу декодировать идентификатор типа данных или же необходимо извлечь из оставшейся битовой строки еще данные. Поэтому вполне логично, что на основе этого значения мы определяем два варианта метода **decode_tag/1**.

```
decode_tag(<<Class:2, Form:1, 2#11111:5, Rest/binary>>) ->
  {TagValue, TagRest} = decode_tag_value(Rest, []),
  {#tag{class = decode_tag_class(Class), form = decode_tag_
form(Form), tag_value = TagValue}, TagRest};
decode_tag(<<Class:2, Form:1, TagValue:5, Rest/binary>>) ->
  {#tag{class = decode_tag_class(Class), form = decode_tag_
form(Form), tag_value = TagValue}, Rest}.
```

Метод **decode_tag_class/1** служит для декодирования значения класса типа данных в соответствующий предопределенный атом. Для декодирования мы используем несколько вариантов функции **decode_tag_class/1**, которые покрывают весь диапазон возможных значений для класса типа данных.

```
decode_tag_class(2#00) -> universal;
decode_tag_class(2#01) -> application;
decode_tag_class(2#10) -> context_specific;
decode_tag_class(2#11) -> private.
```

Метод **decode_tag_form/1** служит для декодирования значения формы типа данных; принципы его работы полностью аналогичны предыдущему методу.

```
decode_tag_form(0) -> primitive;
decode_tag_form(1) -> constructed.
```

Извлечение идентификатора типа данных, который имеет произвольную длину – более сложная задача (простой случай, когда значение идентификатора типа данных содержится в сегменте размером 5 бит, мы рассматриваем отдельно). Как мы помним, при кодировании идентификатора мы преобразуем его значение в битовую строку, состоящую из целого числа сегментов размером 7 бит. После этого каждый 7-битный сегмент преобразуем в октет (8-битный сегмент), добавляя в качестве старшего бита 1 для всех сегментов, кроме последнего, и 0 для последнего сегмента. Поэтому при декодировании мы будем поступать следующим образом: брать очередной октет (8-битный сегмент), извлекать младшие 7 бит и добавлять к списку 7-битных сегментов. Делать эту операцию мы будем до тех пор, пока нам не встретится октет,

Стандарты ASN.1

- » ITU-T Rec. X.680 | ISO/IEC 8824-1. Спецификация на базовую нотацию.
- » ITU-T Rec. X.681 | ISO/IEC 8824-2. Спецификация на информационные объекты.
- » ITU-T Rec. X.682 | ISO/IEC 8824-3. Спецификация на ограничения.
- » ITU-T Rec. X.683 | ISO/IEC 8824-4. Спецификация на параметризацию ASN.1.
- » ITU-T Rec. X.690 | ISO/IEC 8825-1. Спецификация на BER (Basic encoding rules), CER (Canonical encoding rules) и DER (Distinguished encoding rules).
- » ITU-T Rec. X.691 | ISO/IEC 8825-2. Спецификация на PER (Packed encoding rules).
- » ITU-T Rec. X.692 | ISO/IEC 8825-3. Спецификация на ECN (Encoding control notation).
- » ITU-T Rec. X.693 | ISO/IEC 8825-4. Спецификация на XER (XML Encoding rules).
- » ITU-T Rec. X.694 | ISO/IEC 8825-5. Спецификация на отображение на XSD.
- » ITU-T Rec. X.695 | ISO/IEC 8825-6. Спецификация на регистрацию и применение инструкций кодирования PER (Packed encoding rules).

старший бит у которого равен 0. После этого мы меняем порядок сегментов в списке на обратный (т.к. по соображениям производительности мы добавляем сегменты в конец списка), преобразуем список сегментов в битовую строку и извлекаем из полученной битовой строки целое число.

```
decode_tag_value(<<0:1, Segment:7, Rest/binary>>, SegmentList) ->
  TagValueBitstring = list_to_bitstring(lists:reverse([<<Segment:
7>>] ++ SegmentList)),
  BitSize = bit_size(TagValueBitstring),
  <<TagValue:BitSize/integer-big>> = TagValueBitstring,
  {TagValue, Rest};
decode_tag_value(<<1:1, Segment:7, Rest/binary>>, SegmentList) ->
  decode_tag_value(Rest, [<<Segment:7>>] ++ SegmentList).
```

Следующий, не менее необходимый шаг – это декодирование длины (количества октетов), которую занимают данные. Давайте вспомним, как мы кодируем длину, занимаемую данными. Если значение длины больше 128, то для хранения достаточно одного октета (следует отметить, что при этом у октета со значением длины старший бит будет равен 0). Если значение длины больше или равно 128, то первым идет октет, у которого старший бит установлен в 1, а остальные биты содержат количество октетов для хранения длины, после чего идет сама длина (занимающая целое число октетов). Функциональность по декодированию длины реализует функция **decode_length/1**; вполне логично, что она содержит два варианта, покрывающих два возможных случая хранения длины в закодированном виде.

```
decode_length(<<0:1, Length:7, Rest/binary>>) -> {Length, Rest};
decode_length(<<1:1, LengthOctetCount:7, Rest/binary>>) ->
  LengthBitCount = 8 * LengthOctetCount,
  <<Length:LengthBitCount, ParseRest/binary>> = Rest,
  {Length, ParseRest}.
```

Теперь перейдем непосредственно к декодированию объектов Erlang. Начнем с декодирования булевских значений (булевские значения в языке Erlang являются не особым типом, а двумя предопределенными атомами **true** и **false**). Тип данных для булевских значений равен 1 (класс – **universal**, форма – **primitive**, идентификатор – 1), для хранения собственно значения достаточно 1 октета. »

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Стандартные типы данных ASN.1

Стандартные типы данных ASN.1 – это типы данных, описанные в стандартах ASN.1. У этих типов данных класс **universal**; работать с данными стандартных типов должна уметь любая ASN.1-совместимая реализация протокола. К стандартным типам данных относятся следующие:

» **Логические значения (BOOLEAN)**

Принимают два значения – TRUE и FALSE.

» **Значение NULL** Собственно говоря, это не тип, а специальное значение, обрабатываемое и кодируемое специальным, отличным от любых других значений любых типов способом.

» **Целые числа (INTEGER)** Содержит целые числа произвольного размера.

» **Перечисления (ENUMERATED)** Подмножество целых чисел; с точки зрения правил кодирования ASN.1 BER, значения этого типа кодируются совершенно так же, как и целые числа.

» **Действительные числа (REAL)** Содержит действительные числа произвольного размера. С точки зрения правил кодирования ASN.1 BER действительные числа по основанию 10 и по основанию 2 кодируются по-разному. Правила кодирования ASN.1 DER определяют один формат кодирования для действительных чисел – как по основанию 10, так и по основанию 2.

» **Битовые строки (BIT STRING)** Содержат последовательность бит, количество которых не кратно 8.

» **Строки октетов (OCTET STRING)** Содержат последовательность октетов (или, что, то же самое, последовательность бит, количество которых кратно 8).

» **Идентификатор объектов (OBJECT IDENTIFIER и RELATIVE-OID).**

» **Последовательность объектов (SEQUENCE и SEQUENCE OF)** Тип данных для хранения объектов других типов в определенной последовательности. Разница между SEQUENCE и SEQUENCE OF в том, что в последовательностях первого типа допускается хранение данных разных типов, тогда как в последовательностях второго типа допускается хранение данных одного типа. С точки зрения правил кодирования ASN.1 BER, SEQUENCE и SEQUENCE OF одинаковы.

» **Множество объектов (SET и SET OF)** Тип данных для хранения объектов других типов, при этом порядок хранения не определен. Разница между SET и SET OF в том, что во множествах первого типа допускается хранение данных разных типов, тогда как во множествах второго типа допускается хранение данных одного типа. С точки зрения правил кодирования ASN.1 BER SET и SET OF одинаковы.

Если закодированное значение равно 0, то соответствующее булевское значение равно **false**; если нет, то **true**.

```
decode_boolean(<<1:8, 1:8, Value:8, Rest/binary>>, _
DecodeDispatcher) ->
if
Value == 0 -> {false, Rest};
Value /= 0 -> {true, Rest}
end.
```

Еще один тип данных, которые мы хотим научиться декодировать – целые числа. Тип данных для булевских значений равен 2 (класс – **universal**, форма – **primitive**, идентификатор – 2); длина, необходимая для хранения закодированных данных (в отличие от предыдущего случая), может быть любой. Поэтому, прежде чем декодировать данные, мы должны получить длину (при помощи функции **decode_length/1**), после чего сможем декодировать целое число при помощи операции соответствия шаблону (для этого нам нужно знать количество бит, которые занимает целое число в битовой строке).

```
decode_integer(<<2:8, Rest/binary>>, _DecodeDispatcher) ->
{OctetCount, OctetCountRest} = decode_length(Rest),
Length = OctetCount * 8,
<<Number:Length/integer-signed-big, ParseRest/binary>> =
OctetCountRest,
{Number, ParseRest}.
```

Давайте пойдем дальше: рассмотрим, как мы будем декодировать действительные числа. Хранение действительных чисел в за-

кодированном виде достаточно сложно, несмотря на то, что сами значения хранятся в виде строк. Тип данных для действительных чисел равен 9 (класс – **universal**, форма – **primitive**, идентификатор – 9). Если длина закодированных данных равно 0, то это означает, что мы закодировали действительное число 0.0. Если длина закодированных данных равна 1 (т.е. данные занимают один октет), а после длины идет октет со значением **2#01000000**, то это означает, что мы закодировали действительное число +∞ (в языке Erlang для этого мы используем самое большое действительное число $1.7976931348623157 * 10^{308}$). Если длина закодированных данных равна 1 (т.е. один октет), а после длины идет октет со значением **2#01000001**, то это означает, что мы закодировали действительное число -∞ (в языке Erlang для этого мы используем самое маленькое действительное число $-1.7976931348623157 * 10^{308}$). И, наконец, если длина закодированных данных больше 1, то это означает, что нам необходимо декодировать данные дальше.

То, как мы будем декодировать данные дальше, зависит от октета, который идет после закодированной длины (2 старших бита этого октета равны 0, а оставшиеся 6 бит называются **NR**). Если значение **NR** равно **2#000001**, то закодированные данные содержат целое число в строковом представлении. Если значение **NR** равно **2#000010**, то закодированные данные содержат действительное число с фиксированной запятой в строковом представлении. Если значение **NR** равно **2#000011**, то закодированные данные содержат действительное число с плавающей запятой в строковом представлении.

```
decode_real(<<9:8, 0:8, Rest/binary>>, _DecodeDispatcher) ->
{0.0, Rest};
decode_real(<<9:8, 1:8, 2#01000000:8, Rest/binary>>, _
DecodeDispatcher) -> {1.7976931348623157e308, Rest};
decode_real(<<9:8, 1:8, 2#01000001:8, Rest/binary>>, _
DecodeDispatcher) -> {-1.7976931348623157e308, Rest};
decode_real(<<9:8, Rest/binary>>, _DecodeDispatcher) ->
{TotalOctetCount, OctetCountRest} = decode_length(Rest),
OctetCount = TotalOctetCount - 1,
<<2#00:2, NR:6, RealBinary:OctetCount/binary, ParseRest/
binary>> = OctetCountRest,
RealStr = binary_to_list(RealBinary),
case NR of
2#000001 -> {list_to_integer(RealStr) * 1.0, ParseRest};
2#000010 -> {list_to_float(RealStr), ParseRest};
2#000011 -> {list_to_float(RealStr), ParseRest}
end.
```

А мы пойдем дальше: рассмотрим, как декодировать строку октетов (байтовую строку). Тип данных для строки октетов равен 4 (класс – **universal**, форма – **primitive**, идентификатор – 4). После типа данных идет длина, а после нее октеты, составляющие строку октетов. Для их извлечения мы используем операцию соответствия шаблону (для этого нам нужно знать их количество, т.е. длину).

```
decode_octetstring(<<4:8, Rest/binary>>, _DecodeDispatcher) ->
{OctetCount, OctetCountRest} = decode_length(Rest),
<<Octet:OctetCount/binary, ParseRest/binary>> =
OctetCountRest,
{Octet, ParseRest}.
```

Задача, близкая к предыдущей – декодирование битовой строки (строки, в которой количество бит не кратно 8). Тип данных для строки октетов равен 3 (класс – **universal**, форма – **primitive**, идентификатор – 3). После типа данных идет длина, за которой идут данные. Т.к. длина данных – это количество октетов, необходимых для их хранения, а количество бит в битовой строке не кратно 8, то нам необходимо как-то узнать размер остатка битовой строки,

» **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.

который хранится в последнем октете. Для этого первый октет закодированных данных содержит количество неиспользуемых бит в последнем октете, после чего располагается сама битовая строка. Очевидно, что в этом случае длина, требуемая для хранения данных, на 1 больше количества октетов, необходимых для хранения битовой строки.

```
decode_bitstring(<<3:8, Rest/binary>>, _DecodeDispatcher) ->
{OctetCount, OctetCountRest} = decode_length(Rest),
<<UnusedBitCount:8, UnusedBitCountRest/binary>> =
OctetCountRest,
BitstringLength = 8 * (OctetCount - 1) - UnusedBitCount,
<<Bitstring:BitstringLength/bitstring, _
UnusedBits:UnusedBitCount, ParseRest/binary>> =
UnusedBitCountRest,
{Bitstring, ParseRest}.
```

Займемся декодированием составных типов данных. Начнем с декодирования последовательностей (или списков). Для них значение типа данных равняется 48 (класс – **universal**, форма – **constructed**, идентификатор – 16). После декодирования типа данных мы получаем длину закодированной последовательности, после чего строку-октет полученной длины мы декодируем так же, как исходную битовую строку. Именно для этого случая мы и передаем в функции декодирования отдельных типов данных функцию-диспетчер. Декодирование содержимого последовательности осуществляется в функции `decode_sequence_content/3`.

```
decode_sequence(<<0:2, 1:1, 16:5, Rest/binary>>, DecodeDispatcher) ->
{OctetCount, OctetCountRest} = decode_length(Rest),
<<SequenceBinary:OctetCount/binary, SequenceRest/binary>> =
OctetCountRest,
Sequence = decode_sequence_content(SequenceBinary,
DecodeDispatcher, []),
{Sequence, SequenceRest}.
```

Декодирование кортежей осуществляется по тем же самым принципам, что и декодирование последовательностей. Для кортежей значение типа данных равняется **16160 = 2#0011111100100000** (класс – **universal**, форма – **constructed**, идентификатор – 32). Отличие этой функции от предыдущей заключается в том, что после декодирования полученную последовательность (или список) мы преобразуем в кортеж.

```
decode_tuple(<<0:2, 1:1, 2#11111:5, 32:8, Rest/binary>>,
DecodeDispatcher) ->
{OctetCount, OctetCountRest} = decode_length(Rest),
<<SequenceBinary:OctetCount/binary, SequenceRest/binary>> =
OctetCountRest,
Sequence = decode_sequence_content(SequenceBinary,
DecodeDispatcher, []),
{list_to_tuple(Sequence), SequenceRest}.
```

Метод `decode_sequence_content/3` декодирует последовательность октетов, которую мы распознали как последовательность. Для этого данный метод использует передаваемую ему через один из параметров функцию-диспетчер и метод `decode/2`. Может возникнуть следующий вопрос: мы изначально вызываем функцию `decode/2` для декодирования данных, и в результате этого декодирования мы снова вызываем функцию `decode/2`; не приведет ли это к бесконечной рекурсии? Но если хорошо подумать, то можно ответить, что нет, т.к., мы всякий раз вызываем функцию `decode/2` для битовой строки, которая меньше исходной.

```
decode_sequence_content(<<>>, _DecodeDispatcher, ContentList) ->
lists:reverse(ContentList);
decode_sequence_content(Binary, DecodeDispatcher, ContentList) ->
{DecodedElement, DecodeRest} = decode(Binary,
DecodeDispatcher),
decode_sequence_content(DecodeRest, DecodeDispatcher,
[DecodedElement] ++ ContentList).
```

И, наконец, последний метод для декодирования данных работает с закодированными атомами. Для атомов значение типа данных равняется **16161 = 2#0011111100100001** (класс – **universal**, форма – **constructed**, идентификатор – 33). Во всем остальном этот метод тривиален.

```
decode_atom(<<0:2, 0:1, 2#11111:5, 33:8, Rest/binary>>, _DecodeDispatcher) ->
{OctetCount, OctetCountRest} = decode_length(Rest),
<<AtomBinary:OctetCount/binary, ParseRest/binary>> =
OctetCountRest,
{binary_to_atom(AtomBinary, utf8), ParseRest}.
```

Наш пример по кодированию и декодированию объектов Erlang в соответствии с правилами **ASN.1 BER** закончен. Осталось только проверить, что все работает. Если подходить к такой проверке правильно, то необходимо убедиться, что каждый тип поддерживаемых объектов кодируется и декодируется нашими модулями должным образом (данные проверки удобно реализовать при помощи unit-тестов; мы поговорим о unit-тестировании приложений для Erlang в одной из будущих статей). Кроме того, мы помним, что **ASN.1** – это стандарт взаимодействия различных приложений, написанных на разных языках и под разные платформы. Поэтому необходимым шагом проверки будет проверка взаимодействия нашего приложения со сторонним приложением: данные, закодированные нашим приложением, должны быть декодированы сторонним, и наоборот (за некоторыми исключениями, о которых мы поговорим в заключении). Однако из-за ограничения места под статью автор приведет лишь пример, что сложная структура данных Erlang после кодирования и декодирования не меняется; это можно считать неплохим smoke-тестом. Что мы делаем: запускаем среду выполнения Erlang, компилируем модули `asn1_encoder` и `asn1_decoder`, создаем функции-диспетчеры для кодирования и декодирования, после чего кодируем некоторый сложный объект и декодируем его. Если все работает правильно, то после декодирования мы должны получить точно такой же объект, как до кодирования.

```
c(asn1_encoder).
c(asn1_decoder).
Source = {abc, 12, 3.14, [], [{ab, true}, <<1:8>>, <<1:7>>], {}}.
Encoder = asn1_encoder:build([]).
Decoder = asn1_decoder:build([]).
Data = asn1_encoder:encode(Source, Encoder).
{Dest, Rest} = asn1_decoder:decode(Data, Decoder).
```

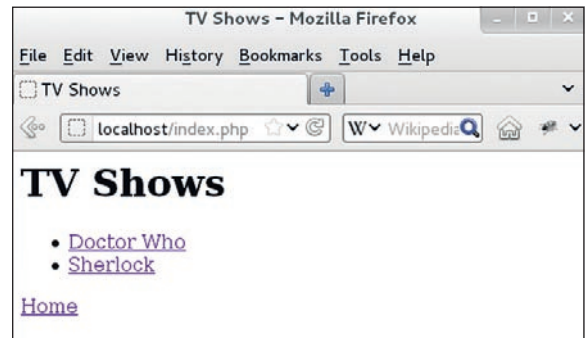
После выполнения всех вышеприведенных команд мы получаем, что переменная `Dest` (объект после кодирования и декодирования) содержит точно такой же объект, что и `Source`, а переменная `Rest` – пустую битовую строку (что логично, ибо у нас не должно остаться не декодированных данных). Можно считать, что наш пример прошел smoke-тест.

Мы закончили большой пример по использованию битовых строк. Про нашу реализацию кодирования и декодирования в соответствии с правилами **ASN.1 BER** можно сказать следующее: мы не реализовали поддержку всех стандартных типов **ASN.1** (мы не поддерживаем, например, идентификаторы, строки, значение NULL) и мы ввели специфичные только для языка Erlang типы (это кортежи и атомы). Что касается первого замечания, то мы исходили из целей построить систему, достаточную для примера, а не для реального использования (и, как кажется автору, это получилось). Что же касается второго замечания, то вполне нормальная практика, когда мы помимо стандартных типов вводим специфичные и известные только для нашего приложения.

На этом, пожалуй, можно поставить точку. Мы завершаем цикл статей, посвященный практикуму по функциональному программированию в языке Erlang. В следующем номере мы приступим к разговору про многозадачность и построение распределенных систем. **LXF**

SQL: Дизайн базы

Для большинства людей база данных подразумевает реляционную базу вроде MySQL или SQLite. А Дейв Кросс создает базу данных про телешоу.



» На главной странице приложения показан список телешоу, содержащихся в базе данных.



Наш эксперт

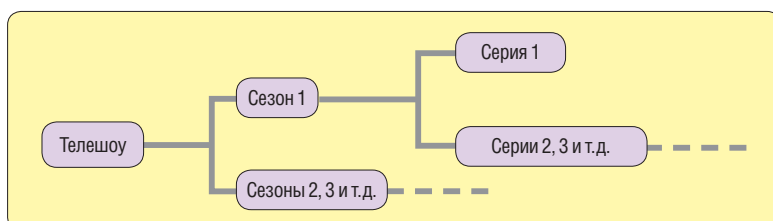
Дэйв Кросс уже не помнит, когда начал пользоваться SQL и создавать базы данных, а прочитанным стандартам ANSI SQL он и счет потерял.

В реляционных базах данных с помощью SQL производится и объявление таблиц, и работа с данными в этих таблицах. Поэтому внутри SQL есть два различных языка – DDL (Data Definition Language – язык определения данных) и DML (Data Manipulation Language – язык управления данными). Сначала познакомимся с DDL.

При создании базы данных полезно начать с перечисления объектов реального мира, сведения о которых нужно хранить в базе. Каждый из таких объектов скорее всего будет представлен таблицей в нашей базе данных. Как обещано, мы будем хранить данные о телешоу, поэтому первой таблицей будет `tv_show`. И под телешоу я имею в виду нечто вроде *Доктора Кто* или *Шерлока*, а не отдельную серию. Поэтому наша следующая таблица – `episode` [серия]. Но в большинстве телешоу серии еще и объединяются в сезоны, вот вам и следующая таблица – `series` [сезоны].

Обрисовем сцену

На этом этапе полезно нарисовать схему нашей базы данных. Мы нарисуем так называемую «схему сущность–связь». На таких схемах таблицы (или сущности) изображаются в виде прямоугольников, а связи между ними – в виде линий:



Обратите внимание, что каждая линия на дальнем конце разветвляется. Это означает, что с этой стороны отношения может быть несколько экземпляров сущности. Так, наша схема говорит, что у телешоу есть несколько сезонов, а в каждом сезоне – несколько серий. Эти отношения «один ко многим» – ключевая идея в проектировании баз данных. Вскоре мы рассмотрим, как проектировать таблицы с такими отношениями. Начнем с команды DDL для создания нашей базы данных. В одной установке MySQL может быть несколько отдельных баз данных, и для каждого нового проекта стоит создавать новую базу данных и нового пользователя, который будет владельцем таблиц в этой базе данных. Для начала нужно подключиться к серверу MySQL от имени суперпользователя-root, но после создания базы данных и нового пользователя административный доступ больше не понадобится. Вот начальная команда для подключения к базе данных:

```
mysql -u root -p
```

В большинстве установок MySQL начальный пароль пользователя root по умолчанию пуст. Если вам он понадобится, можете его изменить.

Команда для создания новой базы данных довольно проста:

```
create database television;
```

Точка с запятой в конце говорит MySQL, что команда закончилась. Если нажать Enter в конце строки, не завершающейся точкой с запятой, MySQL будет ждать ввода дальнейших команд и ничего не выполнит. Таким способом легко набирать сложные команды, состоящие из нескольких строк. Команда для создания нового пользователя менее очевидна:

```
grant all on television.*  
to 'television'@'localhost'  
identified by 'PASSWORD';
```

На самом деле, создание нового пользователя – лишь побочный эффект этой команды. Ее основная цель – дать пользователю права доступа к объектам базы данных. Здесь мы говорим, что хотим дать все права доступа ко всем объектам в базе данных `television (television.*)` пользователю `television`, и он будет подключаться с локального хоста с паролем "PASSWORD". При создании вашего пользователя советую взять другой пароль! Теперь есть надежда подключиться к нашей базе данных MySQL командой

ДАнных



```
mysql -utelevision -Dtelevision -p
```

Параметр **-u** — имя пользователя, **-D** — имя базы данных, а **-p** велит *MySQL* запросить пароль. Пароль можно указать прямо в командной строке, но если вы не уверены, что никто не стоит у вас за спиной, лучше ввести его в ответ на запрос.

Создаем таблицы

Пора создавать таблицы. Но перед этим нужно решить, какие данные для различных сущностей будут храниться. Элемент данных в базе — это столбец. В каждой из наших таблиц их будет совсем немного: например, в таблице **tv_show** их будет всего два. Для каждого столбца нужно указать имя и тип данных, определяющий, какие данные будут там храниться.

```
create table tv_show (
  id integer not null auto_increment,
  title varchar(100) not null,
  primary key(id)
) engine=InnoDB;
```

Хотя это простая таблица, в ее определении нужно кое-что объяснить. В таблице всегда стоит заводить первичный ключ [primary

key]. Это целое число, увеличивающееся на единицу с каждой записью, которое однозначно определяет строку в таблице. Я всегда называю этот столбец **id**. Мы определили его как **not null**, что означает, что в нем всегда должно быть значение; но также указав **auto_increment**, мы позволяем *MySQL* самому заполнять эти значения. Каждый раз при добавлении новой строки в таблицу *MySQL* будет определять следующее неиспользованное значение идентификатора и заносить его в столбец **id**. Затем мы определяем столбец **title** — это единственные данные, которые мы будем хранить в данной таблице. Мы определяем их тип как **varchar(100)**, то есть это строка длиной до 100 символов.

В следующей строке определения таблицы не задается нового столбца: она просто сообщает *MySQL*, какой из наших столбцов является первичным ключом. Наконец, после списка столбцов мы говорим *MySQL*, какой движок базы данных использовать. Это неважно, пока мы не говорим о ссылочной целостности данных. А говорить об этом мы не можем, пока у нас нет хотя бы двух таблиц. Вот наша таблица **series**.

```
create table series (
  id integer not null auto_increment,
  number integer not null,
  name varchar(100),
  tv_show integer not null,
  primary key (id),
  foreign key (tv_show) references tv_show(id)
) engine=InnoDB;
```

Она начинается похоже на таблицу **tv_show**. У нас есть первичный ключ, под названием **id**. Затем идут два столбца данных, **number** и **name**. Мы определили **number** как **not null**, так как у любого сезона есть номер, но **name** может быть пустым (да и будет по умолчанию пустым, если мы ничего не укажем), так как у отдельных сезонов обычно нет собственных названий. Следующий столбец — **tv_show**, и это целое число, которое не может быть пустым. Этот столбец связывает сезон с соответствующим телешоу.

Из нашей схемы «сущность—связь» мы знаем, что телешоу состоит из нескольких сезонов. В нашей модели за это отвечает столбец **tv_show** таблицы **series**, содержащий **id** соответствующего



Щелкнув по сезону, вы получите полную информацию о сезоне и список серий.

Краткий курс истории РБД

Термин «реляционная база данных» вышел из-под пера Тэда Кодда [Ted Codd] в одном из документов, которые он написал, работая в IBM в 1970 году. Этот документ «*A Relational Model of Data for Large Shared Data Banks* [Реляционная модель данных для больших разделяемых банков данных]» все еще доступен на <http://mgnm.at/codd>. Тем, кто не знаком с теорией множеств и реляционной алгеброй, читать его будет нелегко, но интересно, если вы вникаете в историю баз данных и готовы приложить усилия.

За следующее десятилетие Кодд и его команда усовершенствовали свои идеи. В начале восьмидесятых они опубликовали список из 12 правил, которым должна следовать истинно реляционная база данных (его можно прочесть на <http://mgnm.at/12rules>). Инте-

ресно отметить, сколь немногим из этих правил следуют современные реляционные БД. До Кодда базы данных уже существовали, но по большей части были основаны на проприетарных технологиях, и с ними было трудно работать.

И эти изменения произошли не внезапно. В середине восьмидесятых, на моем академическом учебном курсе, SQL изучался как новая технология. Я даже помню, что он назывался языком запросов, подходящим для конечных пользователей. Эта мечта продлилась недолго...

Если вам интересна более теоретическая сторона реляционных баз данных, советую прочесть книги Криса Дейта [Chris Date], в частности, «*SQL and Database Theory* [SQL и теория баз данных]».

Скорая помощь

SQL — сокращение от «Structure Query Language» (язык структурированных запросов). Изначально он был среди многих экспериментальных языков для работы с базами данных, но выжил в итоге один.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

телешоу. Так, если в таблице `tv_show` у нас есть строка для сериала «Доктор Кто» и ее идентификатор равен 10, то в каждой строке в таблице сезонов для «Доктора Кто» в столбце `tv_show` будет стоять 10. Столбец, ссылающийся на родительскую таблицу, называется внешним ключом [foreign key], так как он содержит значения из столбца первичного ключа, но они не являются первичными ключами этой таблицы. Для задания внешних ключей мы добавляем clauses к определению таблицы. В этой таблице мы говорим, что столбец `tv_show` – внешний ключ; он содержит значения столбца `id` таблицы `tv_show`. В таблице допускается всего один первичный ключ, но внешних вполне может быть несколько. Позже мы рассмотрим такой пример.

У внешних ключей есть два важных свойства. Во-первых, поскольку сезон принадлежит телешоу, в каждой строке таблицы `series` обязательно должно быть значение в столбце `tv_show`. Для этого мы принудительно объявляем столбец как непустой. А во-вторых, каждое значение в столбце `tv_show` должно быть действительным, существующим в таблице `tv_show`. Именно это я имел в виду ранее, когда упоминал о ссылочной целостности данных. Если попробовать вставить в таблицу `series` строку со значением поля `tv_show`, которого нет в таблице `tv_show`, вставка должна завершиться неудачно. Точно так же, при попытке удалить строку в таблице `tv_show` удаление должно завершиться неудачно, иначе в таблице `series` останутся строки-сироты.

В ранних версиях *MySQL* не было поддержки внешних ключей – их нельзя было даже задать. Позже такая возможность появилась, но *MySQL* внешние ключи не контролировал, то есть можно было иметь внешние ключи, указывающие на несуществующие записи. С версии 4 в *MySQL* используются жесткие внешние ключи, но только в движке *InnoDB*. В движке по умолчанию *MyISAM* жесткие внешние ключи все-таки не используются, и поэтому мы указываем движок *InnoDB* в определениях таблиц.

Вот определение нашей третьей таблицы, `episode`.

```
create table episode (
  id integer not null auto_increment,
```



Скорая помощь
MySQL – самая популярная открытая реляционная СУБД. Но большая часть данного учебника вполне подойдет и к *SQLite* и *PostgreSQL*.

Подробнее об SQL

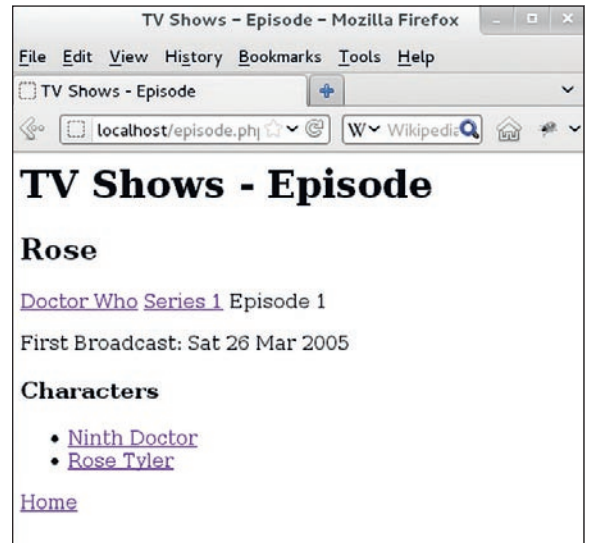
Как я отметил в основной статье, SQL состоит из двух основных частей: языка определения данных (DDL) и языка управления данными (DML). Мы лишь отчасти познакомились с каждым из них; добавлю вам информации. Полный набор команд *MySQL* с описанием можно найти в онлайн-документации на <http://dev.mysql.com/doc>.

Оплот DDL – выражение **create**. На нашем уроке с его помощью мы создавали базы данных и таблицы. Другое частое применение этого выражения – создание индексов. С помощью индексов в базах данных можно производить более быстрый поиск путем сохранения списков мест расположения конкретных фрагментов данных. Они аналогичны указателям в книгах.

Противоположные задачи решает выражение **drop**. С его помощью можно удалить индексы, таблицы и даже базы данных. Помните, что таблицу удалить нельзя, если это нарушит ссылочную целостность данных. Например, нельзя удалить таблицу `tv_shows`, если есть записи в таблице `series`, так как все они будут содержать ссылки на таблицу `tv_shows`. Последняя

команда DDL, которая может вам пригодиться – **alter table**. Она делает именно то, что следует из названия: предоставляет мощный и гибкий синтаксис для изменения таблиц, вместо их удаления и повторного создания. С ее помощью, например, можно добавить или удалить столбец.

DML предназначен для работы с данными в таблицах. Четыре основных процесса управления данными часто описываются с помощью аббревиатуры CRUD, которая означает «**Create, Read, Update и Delete**» [создание, чтение, обновление и удаление]. Мы уже работали с командами **insert** (которая создает данные в таблицах) и **select** (которая считывает данные из таблиц). Две оставшихся команды – **update** и **delete**. С помощью команды **update** можно изменить значения отдельных столбцов (полей) в строке базы данных, а с помощью команды **delete** – удалить строки целиком. Обои команды требуются выражение **where**, которое говорит им, с какими строками работать. Если его не указать, команды будут работать со всеми строками – что в случае с **delete** может вызвать катастрофу.



Щелкнув на названии серии, вы получите подробную информацию о ней и список появляющихся в ней персонажей.

```
number integer not null,
title varchar(100) not null,
first_broadcast datetime,
series integer not null,
primary key (id),
foreign key (series) references series(id)
) engine=InnoDB;
```

Здесь нам незнаком только новый тип данных. Мы определили столбец `first_broadcast` как `datetime`. В *MySQL* предусмотрено множество predefined типов данных, и всегда имеет смысл воспользоваться тем, который больше всего подходит для хранимых данных.

Это первые три таблицы; теперь добавим в них немного данных.

```
insert into tv_show (title) values ('Доктор Кто');
insert into tv_show (title) values ('Шерлок');
insert into series (number, tv_show) values (1, 1);
insert into series (number, tv_show) values (1, 2);
insert into series (number, tv_show) values (2, 1);
insert into series (number, tv_show) values (2, 2);
insert into episode (number, title, first_broadcast, series)
values (1, 'Поза', '2005-03-26', 1);
insert into episode (number, title, first_broadcast, series)
values (2, 'Конец света', '2005-04-02', 1);
insert into episode (number, title, first_broadcast, series)
values (1, 'Этюд в розовых тонах', '2010-07-25', 2);
insert into episode (number, title, first_broadcast, series)
values (2, 'Слепой банкир', '2010-08-01', 2);
insert into episode (number, title, first_broadcast, series)
values (1, 'Новая земля', '2006-04-15', 3);
insert into episode (number, title, first_broadcast, series)
values (2, 'Зуб и коготь', '2006-04-22', 3);
insert into episode (number, title, first_broadcast, series)
values (1, 'Скандал в Бельгравии', '2012-01-01', 4);
insert into episode (number, title, first_broadcast, series)
values (2, 'Собака Баскервильей', '2012-01-08', 4);
```

Теперь информацию из нашей базы данных можно добыть простым SQL-запросом

```
select title from tv_show;
select tv_show.title, series.number as series,
episode.number as episode, episode.title,
date_format(episode.first_broadcast, '%a %e %b %Y')
```



```
from tv_show
join series on series.tv_show = tv_show.id
join episode on episode.series = series.id;
```

С помощью ключевого слова **join** в SQL можно выбрать данные из нескольких таблиц одним запросом. Для этого в запросе надо указать объединяемые таблицы и отношения первичного и вторичного ключей. В первом примере в запросе выше мы связываем таблицу **tv_show** с таблицей **series**, и внешний ключ в таблице **series (series.tv_show)** соответствует первичному ключу в таблице **tv_series (tv_series.id)**.

Этот запрос также демонстрирует возможности типа данных **datetime**. В *MySQL* есть несколько функций для обработки даты и времени. В этом примере вы воспользовались простейшей из них. Функция **date_format** берет значение даты и времени и переформатирует его. Первый аргумент – значение даты и времени, а второй – строка формата, в котором мы хотим их отобразить. Сочетания **%X** в строке формата представляют различные части даты и времени, которые нужно отобразить. Список этих сочетаний приведен в документации по *MySQL* (о том, где ее найти, см. информацию во врезке).

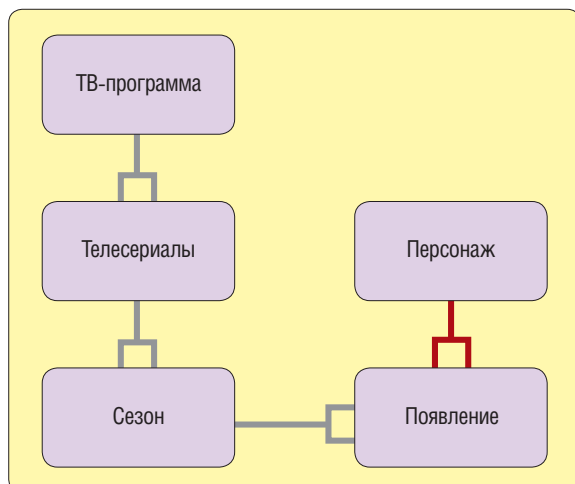
Сейчас мы также можем продемонстрировать мощь ссылочной целостности. Попытавшись удалить строку из таблицы **tv_show**, вы получите сообщение об ошибке, и строка не будет удалена. Причина в том, что для каждой строки в таблице **tv_show** есть строки в таблице **series**, которые на нее ссылаются. Вы не можете удалить строку в **tv_show**, пока не удалите все соответствующие ей сезоны. Аналогично, нельзя удалить строку сезона, пока есть соответствующие ей серии.

Добавляем персонажей

Все это мило, но отношения между нашими таблицами до сих пор были очень простыми: это были отношения «один ко многим». Увы, в мире не всегда все так гладко. Давайте добавим в нашу базу данных информацию о персонажах и их появлении. Какие таблицы нужно создать? И какие внешние ключи выбрать?

Очевидно, нужна таблица с персонажами [**character**]. Но **character** [гж. символ] – также зарезервированное ключевое слово в SQL, поэтому давайте назовем нашу таблицу **person** [личность]. Эта таблица выглядит так:

```
create table person (
  id integer not null auto_increment,
  name varchar(100) not null,
  primary key (id)
) engine=InnoDB;
```



➤ Очередная схема «сущность–связь», показывающая, как дополнительная таблица (Появление) превращает отношение «многие ко многим» в два отношения «один ко многим».

В ней задается только имя персонажа. Мы не соединили таблицу ни с одной из других, потому что не можем. Должна быть связь между записью о персонаже и записью о серии, означающая, что персонаж появляется в серии. Но внешний ключ серии в таблице персонажей поместить нельзя, так как персонаж может появляться в нескольких сериях.

И нельзя поместить внешний ключ персонажа в таблицу с сериями, потому что в одной серии появляется несколько персонажей. Это отношение «многие ко многим». Смоделировать его сложнее, чем отношения «один ко многим», которые встречались нам до настоящего момента.

Но сделать это можно. И ключ к решению задачи приведен несколько абзацев назад. Я сказал, что нам нужна информация о персонажах и об их появлении. Решение – создать отдельную таблицу **appearance** [появления персонажей]. Она выглядит так:

```
create table appearance (
  id integer not null auto_increment,
  person integer not null,
  episode integer not null,
  primary key (id),
  foreign key (person) references person(id),
  foreign key (episode) references episode(id)
) engine=InnoDB;
```

Ключевая фигура

Вся эта таблица состоит из ключей. Это первичный ключ и два внешних ключа – к таблице с персонажами и к таблице с сериями. Каждый раз, когда мы знаем, что некий персонаж появился в какой-то серии, мы можем добавить строку в эту таблицу.

И если у нас есть **episode id** [идентификатор серии], мы сможем получить список всех персонажей, появляющейся в этой серии; и наоборот – по **person id** [идентификатору персонажа] можно получить список всех серий, где он появляется. Вот немного данных для наших новых таблиц:

```
insert into person (name) values ('Девятый Доктор');
insert into person (name) values ('Десятый Доктор');
insert into person (name) values ('Роз Тайлер');
insert into appearance (person, episode) values (1, 1);
insert into appearance (person, episode) values (3, 1);
insert into appearance (person, episode) values (1, 2);
insert into appearance (person, episode) values (3, 2);
insert into appearance (person, episode) values (2, 5);
insert into appearance (person, episode) values (3, 5);
insert into appearance (person, episode) values (2, 6);
insert into appearance (person, episode) values (3, 6);
```

И вот вам несколько запросов, демонстрирующих, как таблица **appearance** работает в обоих направлениях.

```
select episode.title
from episode
join appearance on episode.id = appearance.episode
join person on person.id = appearance.person
where person.name = 'Девятый Доктор';
select person.name
from episode
join appearance on episode.id = appearance.episode
join person on person.id = appearance.person
where episode.title = 'Новая земля';
```

Обновленная схема базы данных приведена слева. На ней ясно видно, что таблица **appearance** является связующим звеном между таблицами **person** и **episode**.

Это все, о чем мы можем узнать в нашем простом руководстве по проектированию баз данных. Я написал простое PHP-приложение, демонстрирующее работу с этой базой данных. Его код можно найти на DVD, а несколько снимков экрана показаны по соседству. **LXF**

Скорая помощь

Две хорошие книги об SQL – *Learning SQL* [Изучаем SQL] Алана Боуля [Alan Beaulieu] и *SQL For Smarties* [SQL для умников] Джо Селко [Joe Celko].

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 Linux для преподавателей
- 2 Разрешение монитора
- 3 Печать меток в OpenOffice
- 4 Установку Mint в VirtualBox
- 5 Прерывистое движение мыши
- 6 Изменения в поведении рабочего стола

1 Лунное серебро

В Я преподаю в университете и хочу познакомить студентов с миром Linux. Однако чтобы студенты могли бы полностью уйти из Windows-окружения, надо решить некоторые вопросы.

Первый из них — доступ к сервисам, таким как Citrix, через web-браузер. У них должна быть возможность доступа к приложениям Silverlight и работы с Exchange Server. Выполняемая ими работа охватывает разнообразные задачи по web-программированию (HTML/CSS/PHP), исследования и компиляции программ и работы с сетью — ну, и простые текстовые процессоры.

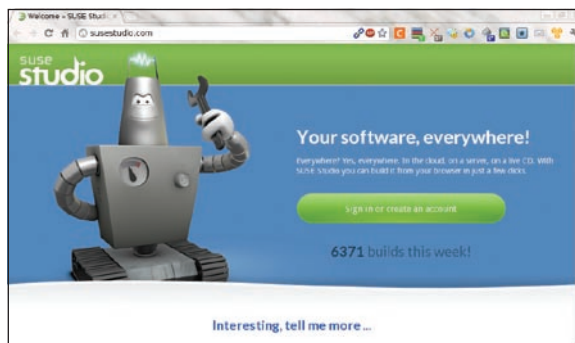
Каким дистрибутивом мне воспользоваться?

Мартин Ротуэлл [Martin Rothwell]

Ваш главный камень преткновения — Silverlight, проприетарная технология Microsoft. Существует открытая реализация этой технологии — Moonlight, но она отстает на целую версию. Последняя версия Moonlight (с не слишком уместным номером 4) поддерживает Silverlight 3, а поддержка Silverlight 4 ограничена. Все зависит от того, какая версия Silverlight нужна серверу Citrix. Модуль расширения для браузера в большинстве дистрибутивов устанавливается через менеджер пакетов либо зайдя на сайт <http://www.go-mono.com/moonlight>. Установите его в тестовой системе и проверьте, получите ли вы доступ к Citrix, а затем двигайтесь дальше.

Следующее препятствие — поддержка Exchange, и самый простой вариант — воспользоваться Evolution: в нем, наверное, самая зрелая поддержка Exchange среди почтовых клиентов Linux. Хотя годятся и другие программы.

Если Moonlight заработает, решение воспользоваться Evolution выведет вас на рабочий стол Gnome. По существу, ваши потребности удовле-



» SUSE Studio поможет создать собственный установочный диск в браузере.

творит почти любой дистрибутив. Если вам нужна конкретная рекомендация, я бы посоветовал openSUSE: у него тесная связь с проектами Mono и Moonlight (некоторые, конечно, считают это поводом избегать продуктов из лагеря SUSE).

Другая причина выбрать openSUSE — программа SUSE Studio (<http://susestudio.com>) для создания собственных дистрибутивов. С ней можно собрать дистрибутив openSUSE, содержащий лишь то, что вам необходимо, и затем создать ISO-образ Live CD или установочного диска — с него вы сможете установить один и тот же дистрибутив заданной конфигурации на все компьютеры.

Все остальное вами упомянутое — стандартные задачи, хорошо выполняющиеся в любом дистрибутиве, хотя собственный установочный диск гарантирует вам наличие всех необходимых программ, в том числе офисных пакетов, языков программирования и среды разработки; это избавит вас от необходимости устанавливать дополнительный набор пакетов после каждой установки, как пришлось бы делать со стандартным дистрибутивом. Вы даже можете оказаться сядугой и удалить игры, поставляемые со многими дистрибутивами.

2 Исчезающее меню

В На моем компьютере установлены Windows Vista и Debian 6. Видеокарта — от Nvidia. Я несколько раз пробовал установить еще один дистрибутив с дисков журнала, и каждый раз после перезагрузки компьютера вместо загрузочного меню появлялся прыгающий прямоугольник с указанием «Пожалуйста, измените разрешение на 1280×1024@69Гц [Please change the resolution to 1280×1024@69HZ]». Через какое-то время это прекращается и загружается новый дистрибутив, но ни в Debian, ни в Vista мне не войти. По-моему, проблема в меню загрузки: я не смог ее выявить

Наши эксперты

» Мы подыщем эксперта на любой ваш вопрос. От установки и проблем с модемом до администрирования сети — ответ найдется! Просто напишите нам бумажное или электронное письмо, и о нем позаботятся.



Нейл Ботвик

Побыв владельцем ISP и редактором дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс

Майк был одним из создателей прототипа LXF — Linux Answers. Его специальности — программирование, оконные менеджеры, скрипты инициализации и SNES.



Джонатан Робертс

Джонатан столько отлынивал от занятий ради установки, настройки и взлома серверов, что теперь он эксперт.



Грэм Моррисон

Когда Грэм не обозревает кучи программного обеспечения и не халтурит с MythTV, он готов к ответам насчет любого оборудования и проблем виртуализации.



Валентин Синецын

В редкие свободные минуты Валентин обычно запускает mcedit, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема — настольный Linux.



Юлия Дронова

Если компьютер у Юлии не занят выполнением команды emerge, она спешит применить его для модерирования www.unixforum.org.

Куда посылать вопросы

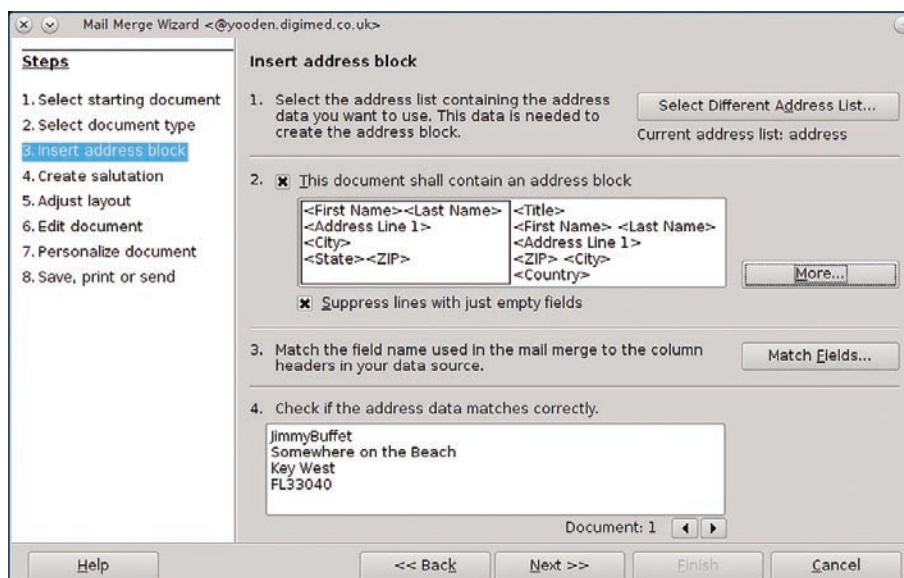
Пишите нам по адресу answers@linuxformat.ru или спрашивайте на форуме: www.linuxformat.ru

обычными методами chroot, и это объяснимо, если моя теория верна.

Случайно я наткнулся на ссылку на файл /etc/default/grub, где решил изменить строку #GRUB_GFXMODE=640x480 на GRUB_GFXMODE=1280x1024. Это помогло, и теперь я могу загрузиться в Vista, в Debian или в новую систему. Все хорошо, но в чем же дело? Исходная строка была закомментирована, и я не могу понять, откуда мой компьютер получал некорректное разрешение?

Брайан Слингер [Brian Slinger]

И старый Grub, и Grub2 по умолчанию используют разрешение 640x480. Это самый стандартный из стандартных режимов, который, как ожидается, поддержит любое устройство. В вашем случае Grub пытается применить разрешение, не поддерживаемое монитором – монитор либо действительно не поддерживает 640x480, либо предоставляет такие данные, которые заставляют Grub поверить, что монитор поддерживает другое разрешение. Когда монитор не может безопасно отобразить какое-то разрешение, он отображает предупреждение, которое вы видели. Это продолжается до тех пор, пока таймер Grub не досчитает до нуля и не будет загружена система по умолчанию (обычно первый пункт меню). Строка GFXMODE по умолчанию закомментирована – это стандартная практика для файлов настройки, чтобы было показано используемое значение по умолчанию, нечто вроде само-документации.



Чтобы в первый раз настроить автоматическое составление меток в LibreOffice, придется немного повозиться, а потом все будет просто.

Вам нужно попробовать GRUB_GFXMODE=640x480, чтобы заставить Grub использовать разрешение 640x480. Если ничего по-прежнему не отображается, то ваш монитор не поддерживает это разрешение. Кстати, видите ли вы загрузочные меню LXF DVD, ведь они используют 640x480? Если монитор отказывается поддерживать 640x480, загляните в его руководство и посмотрите, можно ли включить другие разрешения, кроме предпочитаемого 1280x1024.

Настройки меню Grub можно редактировать на лету, но вам это не пригодится, потому что так вы не увидите, что делаете. Если вы не можете заставить свой монитор работать на 640x480, то единственный вариант – явно указать Grub разрешение, с которым ему работать, так, как вы это уже сделали. Список разрешений, с которыми, по мнению Grub, может работать ваша система, можно просмотреть, нажав с в меню Grub для перехода в командную строку Grub и выполнив в ней команду vbeinfo.



Коротко про...

Screen

Освоив командную строку Linux, вы все охотнее будете стремиться в терминал. Терминал со вкладками xterm позволяет работать сразу в нескольких сеансах, но при удаленном подключении к рабочему столу или в виртуальной консоли им особо не воспользуешься. Несколько сеансов в одном терминале позволит запустить Screen, оконный менеджер для терминалов, и это только начало.

Наберите screen в терминале – и вы уже попали в другой. Наберите ls -l для вывода содержимого текущего каталога, затем нажмите Ctrl+a+c, и все исчезло. Перейдите в другой каталог и снова выполните ls, нажмите Ctrl+a+n – и вы снова видите исходный список. Ctrl+a+c создает новый терминал, оставляя старый нетронутым; Ctrl+a+n переключает между ними. Теперь нажмите Ctrl+a+d для выхода из Screen и возврата в исходную оболочку, откройте другой xterm и наберите screen -r. Вы вернулись в исходный сеанс Screen. Screen прекрасно работает по SSH. Например, можно подключиться по SSH к домашнему компьютеру с работы, за-

пустить Screen, загрузить новую версию какой-то программы, запустить компиляцию и выйти из Screen. Для проверки, как идут дела, подключитесь к своему сеансу Screen командной screen -r с работы или из дома. Можно разделить монитор комбинацией клавиш Ctrl+a+s, переключиться в новую область с помощью Ctrl+a+TAB и нажать Ctrl+a+n (или c) для отображения другого терминала Screen.



Компилируйте программу в верхней части Screen, а в нижней зайдите на чат IRC.

3 Веселых вам рассылок

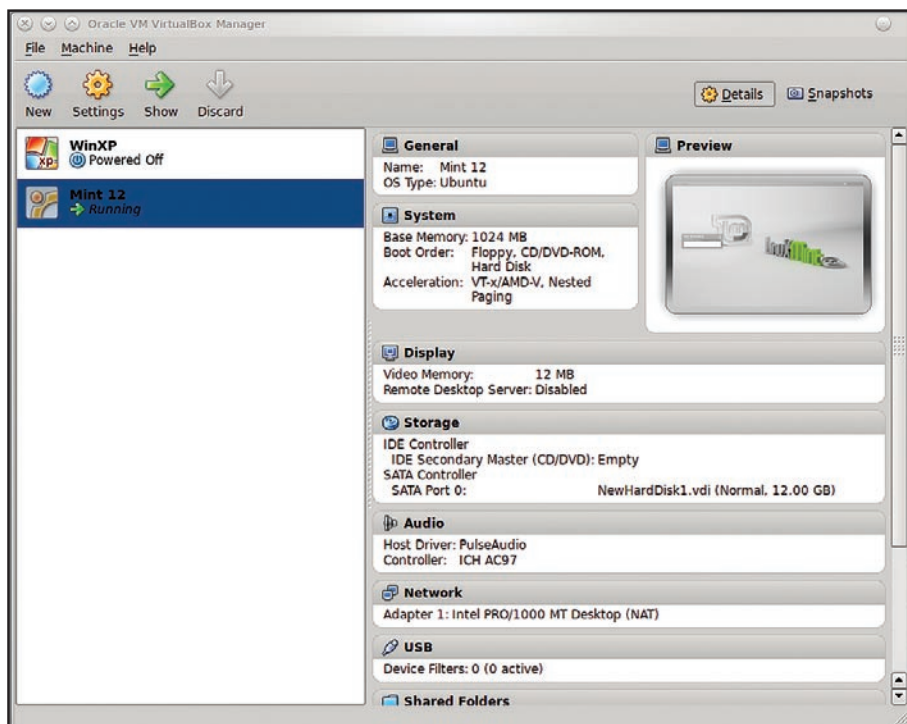
Моя ежегодная забота декабря – печать адресов рождественских открыток. Несколько лет назад я создал таблицу в Microsoft Excel и ссылку в Microsoft Word для автоматического их составления под распечатку.

Затем я обновил компьютер, установив Ubuntu и OpenOffice.org, и теперь не могу их напечатать. Я не понимаю, как напечатать метки из таблицы на листе, где на каждое поле приходится по одному адресу. В Word это просто, а в OpenOffice.org – пугающе сложно, а может, и вовсе невозможно. Хотел бы услышать ваше мнение о том, можно ли это сделать.

Майк Райдер [Mike Ryder]

Сделать это можно, хотя на первый раз и придется повозиться. Нужно отметить, что все это тестировалось не в OpenOffice.org, а в LibreOffice, так как он установлен на всех моих компьютерах. Для OpenOffice.org процедура должна быть похожей, если не в точности такой же. Ubuntu, как и большинство дистрибутивов, переключается на LibreOffice, поэтому описание действий для него будет интересно большинству пользователей – да и вы к следующему Рождеству тоже, вероятно, перейдете на LibreOffice.

Для начала нужно добавить вашу таблицу в виде базы данных, выбрав пункт меню File > New >



► В *VirtualBox* удобно пробовать новые дистрибутивы, но иногда бывает лучше загрузиться с Live CD на реальном «железе».

Database [Файл > Создать > База данных]. В появившемся окне выберите **Connect to an Existing Database** [Подключиться к существующей базе данных] и установите ее тип в Spreadsheet [Таблица]. Нажмите Next [Далее] и выберите файл с таблицей. Нажмите Next [Далее], убедитесь, что выбрана опция регистрации базы данных, и нажмите Finish [Готово]. Укажите подходящее имя для базы данных и нажмите Save [Сохранить]. Это скучная часть нашего труда, и к ней мы больше не вернемся.

Теперь выберите пункт меню File > New > Labels [Файл > Создать > Поля], и откроется окно Label Setup [Настройки полей]. Выберите свежесозданную базу данных и задайте имя таблицы как имя листа с адресами (обычно «Sheet 1»). Выберите тип используемых полей в нижней части окна, затем задайте расположение полей, выбрав поля базы данных (столбцы таблицы), которые нужно вставить. Расположение полей задается на вкладке Format [Формат], а на вкладке Options [Параметры] выберите галочку Synchronise Contents [Синхронизировать содержимое] – это пригодится нам позже. Нажмите New Document [Новый документ], и создастся файл, содержащий всю заданную вами информацию. Решив изменить его, измените только первое поле на странице и нажмите Synchronise Labels [Синхронизировать метки], чтобы изменения применились ко всем полям. Сохраните этот документ как шаблон на все года. Чтобы им воспользоваться, выберите пункт меню Tools > Mail Merge Wizard [Утилиты > Мастер составления писем], укажите, что нужно использовать текущий документ, измените нужные параметры, а затем распечатайте или сохраните документ. На шаге 3 отметьте галочку This Document Shall Contain an Address Block [Этот документ дол-

жен содержать блок адреса] – другие настройки можно не менять. Последующую настройку шаблона можно выполнить нажав кнопку More [Далее], затем Edit [Изменить].

Бывает, что *LibreOffice* печатает чистую страницу между двумя настоящими. Если это происходит, снимите галочку Print Automatically Inserted Blank Pages [Печатать автоматически вставленные чистые страницы] в окне Print [Печать]. Вам может показаться, что хлопот слишком много, и, пожалуй, так и есть, но делается это всего один раз. Затем останется только загружать свой документ и запускать мастер составления писем.

4 Mint зависает

В Мне 82 года, и я потратил несколько часов в безуспешных попытках установить Mint 12 в Oracle VirtualBox с диска LXF154. Я вижу серое окно с надписью «Linux Mint» по диагонали, а затем все, похоже, зависает. Я запускаю VirtualBox в Windows 7 Home. Ubuntu 11.10 вроде установилась нормально.

Остин Майялл [Austin Myall]

○ Непонятно, в какой момент возникает проблема. При выборе Mint в меню загрузки Linux Format появляется серое окно с надписью «Linux Mint», и начинается 10-секундный отсчет до начала загрузки. Все зависает именно здесь, или процесс загрузки начинается, и все зависает на рабочем столе?

Если все зависает в первом окне, то процесс загрузки чем-то прерван. Первое, что стоит попробовать – загрузиться с диска на другом компьютере (не на виртуальной машине), если есть такая возможность. Устанавливать ничего не нужно – просто проверьте, что система загружается. Это

позволит исключить ошибки или повреждения диска (если он не загружается на нескольких компьютерах, свяжитесь с нами для его замены). При появлении начального экрана с таймером обратного отсчета нажмите любую клавишу для отображения загрузочного меню Mint и выберите второй вариант – Compatibility mode (режим совместимости). Если в нем система тоже не загружается, повторите эти действия, но на этот раз нажмите Tab при появлении загрузочного экрана Mint для отображения параметров загрузки и удалите параметры splash и quiet клавишей Backspace.

Нажмите Enter, и система продолжит загружаться без экрана-заставки, который обычно скрывает загрузочные сообщения, и вы сможете увидеть, где останавливается загрузка и с каким сообщением.

Пока вы не увидите это сообщение, невозможно точно определить проблему и найти решение, но добавление acpi=off к параметрам загрузки, из которых вы удалили quiet и splash, ликвидирует достаточное количество проблем.

Если загрузка доходит до рабочего стола, который также серый с надписью «Linux Mint» в середине, но ни иконок, ни меню не появляется, тоже можно попробовать acpi=off. Этот способ решает множество проблем с аппаратными несовместимостями, а при запуске в VirtualBox устройство по сути вдвое больше – реальный компьютер и виртуальный, который работает поверх него.

Еще один вариант – noapic, который не имеет ничего общего с ACPI, несмотря на похожее написание. Можете попробовать добавить и этот параметр к опциям загрузки. Он не имеет аргумента on или off, просто добавьте слово noapic.

5 Хромая мышь

В У меня старый ноутбук Medion MIM 2080 с процессором Intel Celeron 1300 МГц и 1,2 ГБ оперативной памяти. Я настроил конфигурацию с двойной загрузкой (вторая система – Windows XP Home). Я загрузился в Ubuntu 11.10 (с Gnome 3) с LXF DVD155, но мышь движется так, как будто у меня тяжелая форма артрита рук – медленно и рывками. В Windows она работает не хуже, чем раньше.

У меня была та же проблема с мышью, когда я попробовал Mint с LXF DVD154. С более ранней серией Ubuntu 10 мышь работала нормально. Я пробовал изменять чувствительность и скорость мыши, но лучше не становится. Я также, хватаясь за все соломинки, попытался поменять мышь на другую.

Я знаю, что вам нужен максимум информации, но здесь ее слишком мало – что еще сообщить? На другом ноутбуке Aspire стоит Ubuntu 11.10 (Unity) с LXF DVD152 (также с двойной загрузкой с Windows XP Home) – и мышь на нем работает отлично.

Роберт Макдональд [Robert Macdonald]

○ Похоже, это ошибка ядра, которая относится скорее к контроллеру, а не к самой мыши, поэтому замена мыши на другую

и не помогла. Есть несколько способов решения проблемы. Первый – попробовать другую версию ядра: проверьте наличие новой версии в менеджере обновлений и установите ее, затем перезагрузите систему и посмотрите, сохранилась ли проблема. Если проблема остается, попробуйте другой вариант – отключить ACPI при загрузке. Для этого выберите пункт меню для Mint или Ubuntu в загрузочном меню Grub (если меню не появляется, удерживайте Shift при загрузке), нажмите e для изменения пункта меню и добавьте

```
pnpcsr=off
```

в конец строки конфигурации ядра. Нажмите **Ctrl+X** для загрузки с этой строкой конфигурации. Если это не помогло, попробуйте более экстремальный вариант **acpi=off**. Как только мышь заработает, откройте `/etc/default/grub` с правами администратора и добавьте работающую настройку в `GRUB_CMDLINE_LINUX`. Обычно она пуста, поэтому измените ее с

```
GRUB_CMDLINE_LINUX=""
```

на

```
GRUB_CMDLINE_LINUX="pnpcsr=off"
```

Если она не пуста и в ней есть другие настройки, добавьте свою в конец через пробел, следующим образом:

```
GRUB_CMDLINE_LINUX="setting1 setting2 pnpcsr=off"
```

Наконец, сгенерируйте новый файл настройки Grub командой

```
sudo update-grub
```

Когда в менеджере обновлений появится новая версия ядра, попробуйте удалить настрой-

ку в меню загрузки и из файла `/etc/default/grub` и проверьте, будет ли мышь работать без нее.

5 Подвижки рабочего стола

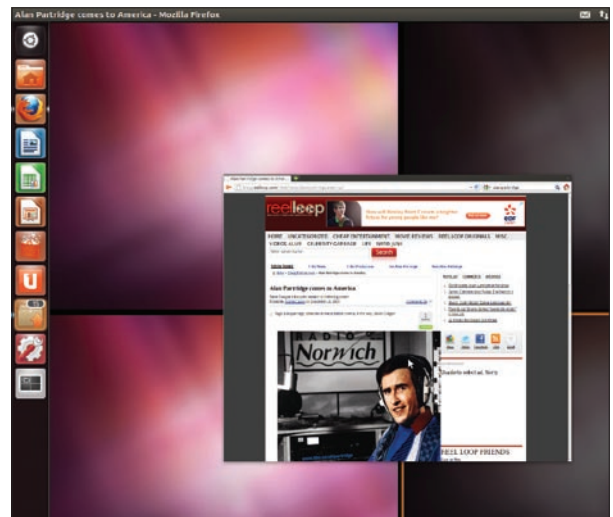
В одних дистрибутивах Ubuntu можно взять окно и перетащить его на следующий рабочий стол, а в других — нельзя. Я не знаю, как это правильно называется, и не могу поискать это в Google. Не могли бы вы объяснить, как оно работает и что нужно сделать, чтобы оно заработало?

Майк Райдер [Mike Ryder]

О Это одно из якобы «улучшений» в Unity и Gnome 3 – окна уже

нельзя перетаскивать между рабочими столами, как раньше. Так что это работает в более старых версиях Ubuntu, но не в нескольких последних релизах. Переключение с Unity на Gnome не поможет. Окна все еще можно перемещать, щелкнув правой кнопкой мыши на заголовке и выбрав Move to Workspace Right/Left [Переместить на рабочий стол справа/слева] или Move to Another Workspace [Переместить на другой рабочий стол].

Существуют и другие варианты, включающие переход в расширенный режим [Expose mode], открытие переключателя рабочих столов [Work-



» Ага! Чтобы перетащить окна между виртуальными рабочими столами, воспользуйтесь переключателем рабочих столов.

space Switcher], нажатие комбинации клавиш и произнесение волшебного слова перед перемещением окна, но заголовок окна – самый простой способ, пока это поведение не вернулось как обычное.

Кстати говоря, можно отправить сообщение об ошибке и попросить, чтобы это поведение было восстановлено, и избежать самодовольных друзей, пользующихся KDE, с болезнью роста, вылившейся в «новую парадигму рабочего стола», и сохранить возможность перетаскивать окна между рабочими столами. **LXF**



Часто задаваемые вопросы

WPA

» **WPA – более надежный наследник WEP. ОК, спасибо за лаконичное объяснение, но ЧТО это?**

WPA (Wi-Fi Protected Access – защищенный доступ по Wi-Fi) – система, гарантирующая приватность данных, передаваемых по беспроводным сетям.

» **Нужно ли мне это?**

Если вы живете не в пещере и не в экранированном бункере, то почти наверняка да. В отличие от традиционной проводной сети, в которой ваши данные не покидают проводов, соединяющих компьютеры, беспроводная сеть распространяет их на сотни метров. Это значит, что любой, у кого есть ноутбук и сетевая карта для беспроводной сети, может легко прочесть ваши данные.

» **У меня большой дом, ноутбук даже не подключен к сети в остальных комнатах, поэтому я в безопасности, так ведь?**

Нет! Хотя, начиная с определенного расстояния, ваш ноутбук может не подключиться к точке доступа, могут обнаружиться другие устройства, которые смогут. Прикрепив к беспроводной сетевой карте антенну, вы получите гораздо больший диапазон. В этом может не быть ничего страшного, если вы пользуетесь Wi-Fi только дома для выхода в Интернет с ноутбука. Ну, а если ваша компания пользуется сетью Wi-Fi со слабой защитой (или без нее), а кто-то в припаркованном снаружи автомобиле считывает все конфиденциальные данные?

» **Раз уж об этом зашла речь – что такое WEP?**

WEP – Wired Equivalent Privacy (приватность, эквивалентная проводной сети). Это была предыдущая попытка

защиты беспроводных сетей. Хотя это лучше, чем полное отсутствие защиты, взломать ее вовсе не трудно. Она может спасти вас от соседей, которые случайно подключатся к вашей точке доступа вместо своей собственной, но не спасет от того, у кого есть намерение взломать ваше соединение.

» **Какие устройства необходимы для WPA?**

Все новые беспроводные сетевые карты и точки доступа поддерживают WPA и WEP. Если у вас более старое устройство, может оказаться, что оно поддерживает только WEP.

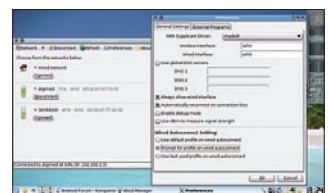
» **Как насчет ПО?**

Вам нужна поддержка WPA на двух уровнях – на уровне драйверов устройства и в отдельной программе, реализующей WPA. Самая популярная из таких программ – `wpa_supplicant`, она должна быть в вашем дистрибутиве. Если ее там нет,

загрузите ее с сайта http://hostap.epitest.fi/wpa_supplicant.

» **Наверное, чтобы воспользоваться WPA в Linux, придется прочесть ман-страницы и HOWTO и отредактировать файлы настройки?**

Можете так и сделать, если жизнь в Vim прибавляет вам ярких эмоций. Но, к счастью, есть и графические утилиты, способные упростить жизнь. Самая продвинутая из них, пожалуй, `Network Manager` в Gnome (<http://www.gnome.org/projects/NetworkManager/>).



» **Network Manager – не единственный возможный вариант, wicd тоже хорош.**

LXF HotPicks



Ник Вейч

Ника однажды укусила радиоактивный менеджер скачиваний, и с тех пор у него болезненная страсть к экспериментам с установкой программ. И дурное пищеварение.

DiffImage » Panther » GSLShell » Smuxi » moneyGuru » IP-Link » haveged
» Scorched » Bitfighter » PedSim » Evince

Инструмент для работы с изображениями

DiffImage

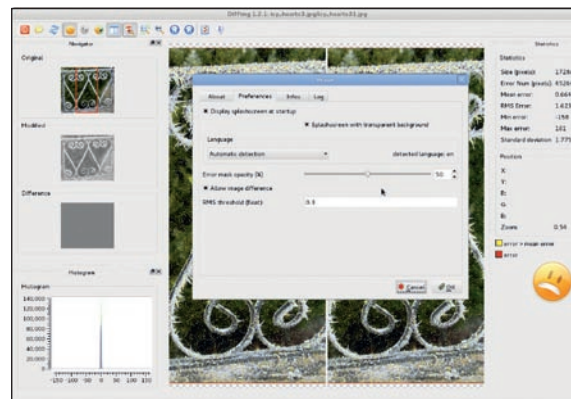
Версия 1.21 Сайт <http://bit.ly/zoeBwC>

Даже крайним тугодумам трудно будет не понять, в чем смысл этой программы: она находит различия в двух файлах изображений. А зачем нужен подобный инструмент? Кроме того, что ее можно натравить на задачи типа «найдите одиннадцать отличий», она предоставляет весьма действенный способ выявить эффекты от различных типов сжатия и определить, где файл был

отредактирован и в какой его части были внесены изменения.

Вообще-то она идет и дальше: здесь есть два уровня рассогласования, выделяемых разными цветами. Желтые области указывают на незначительные изменения, возникающие в результате простой декомпрессии файла или применения легкого размывания, которое в основном сохраняет исходные цвета. Красные области пока-

зывают, где отдельные куски были вырезаны или сильно изменены, как это происходит при соединении двух изображений или силь-



» Чтобы облегчить просмотр изображений, настройте уровень прозрачности через меню Preferences.

«Компиляции проще не бывало в HotPicks с первого выпуска.»

ном сжатии. Если второе изображение сплошь покрыто красным, проверьте, одинаково ли они кадрированы.

Без ограничений, впрочем, не обошлось. Отсутствуют примерное совпадение, вычисление смещения и тому подобное – ваши два изображения должны быть одного размера, да еще и выровнены; если это не так, практически все квалифицируется как различие.

Степень различия также отображается на панели Statistics [Статистика] и в удобной гистограмме внизу слева в основном окне. Так что можно буквально с одного взгляда оценить, имеете ли вы дело с обычными отклонениями вследствие своего рода износа файла (нескольких загрузок и сохранений) или файл претерпел серьезное редактирование (разброс графика будет шире).

Исходник, поставляемый для этой утилиты, несколько нетипичен: он кроссплатформенный, и его надо специально конвертировать под Linux (LF, не CR) перед компиляцией. Простые инструкции, как это сделать, вы найдете в приложенном **INSTALL.txt** (есть даже скрипт вам на подмогу), и было бы справедливо отметить, что помимо этого осложнения, компиляции проще не бывало в HotPicks практически с самого первого выпуска. При наличии *Qt4* и *qtake* все пройдет на ура.

Исследуем интерфейс DiffImage

Выберите изображения

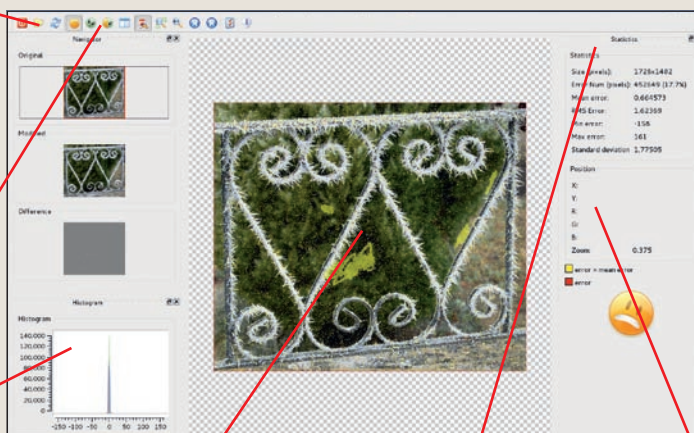
Откройте пару изображений для сравнения с помощью этой кнопки или через меню.

Найдите отличия

Переключайте режим просмотра либо изображений, либо различающихся их частей.

Удобная гистограмма

Гистограмма показывает ошибки и степень отклонений.



Изменение режима просмотра

Просматривайте изображения, расположив их рядом или наложив друг на друга.

Сообщения об ошибках

Статистика отображает подробную информацию об отклонениях изображений.

Добавленная ценность

Этот параметр отображает координаты и показатели RGB для пикселя под курсором.

Транскодер

Panther

Версия 0.9 Сайт <http://sankha93.github.com/Panther>

Создается впечатление, что файлов в нужном человеку формате в жизни не встретишь. Это либо файлы OGG, не воспроизводимые на вашем портативном устройстве, либо MOV-видео, которое не воспроизводится вообще нигде, кроме Mac.

Необходимость в изобилии форматов действительно есть, ведь разные форматы файлов отвечают разным потребностям (качество, размер, простота воспроизведения, пропускная способность и т. д.), но почему бы программам не обрабатывать все эти разнообразные типы? Они, однако, этого не делают, и пользователь остается перед лицом малопривлекательной необходимости конвертирования одного формата в другой.

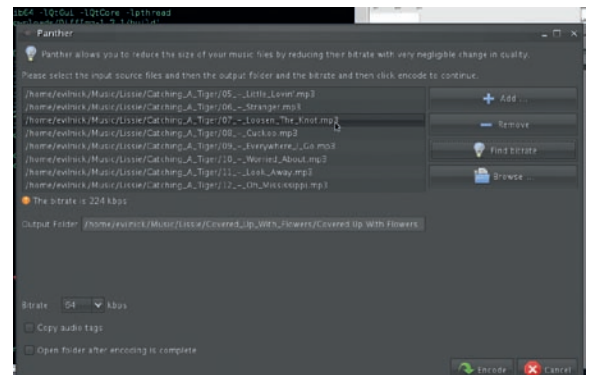
На среднестатистической машине с Linux есть десятки утилит командной строки, которые смогут – самостоятельно или притянув другие инструменты – превратить почти любой формат в поч-

ти любой другой формат. Единственная проблема в том, что у большинства не хватает терпения, чтобы продрасть сквозь 12 map-страниц и выяснить все о секретных ключах команды.

А вот *Panther* предлагает иной способ. Его графический интерфейс на Java позволяет выбрать тип файла и затем за ряд шагов проводит вас по всем соответствующим опциям в дружелюбном к пользователю мастере. Найденные в результате утилиты исполняются, и вы мигом получаете желаемый результат.

Несмотря на применение Java (так и вижу обвал сердитых сообщений электронной почты), интерфейсу пользователя было уделено большое внимание,

«Интерфейс пользователя опрятен, понятен и ничуть не уродлив.»



► *Panther* — для тех, кто любит простоту во всем.

и он вышел опрятным, понятным и ничуть не уродливым. Правда, в некоторых местах имеется тенденция не открывать окно приложения на ширину, достаточную для показа всех имеющихся опций, но это не самое принципиальное. В руках эксперта инструменты командной строки могут справиться с этой работой вдесятеро быстрее, но зато *Panther* — отличная опция для тех, кто немного растерялся и просто хочет, чтобы все работало.

Программа идет в виде файла JAR, и нужно только магическое заклинание

```
java -jar panther.jar
```

чтобы оно заработало.

Итак, удачного вам преобразования форматов.

Числовая оболочка

GSLShell

Версия 2.1.0 Сайт www.nongnu.org/gsl-shell

Числа. Абстрактные и бесконечные. Многие в наши дни считают, что математика — это не просто модель, объясняющая, почему все именно так, а не иначе, но, вполне возможно, связующая ткань самой реальности. И, в погоне за слиянием с этой квинтэссенцией силы творения, мы обращаемся к... электронной таблице?!

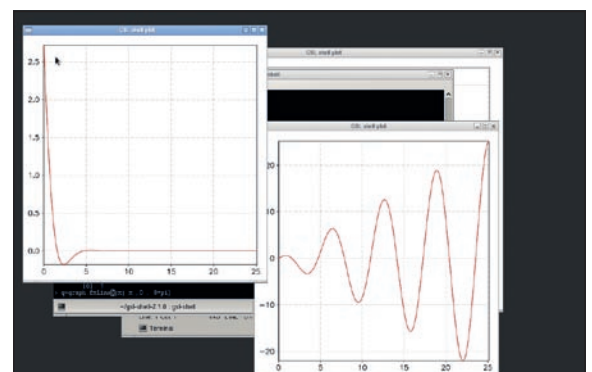
Завораживающую красоту чисел гораздо лучше постигать в среде интерактивной оболочки, каковой и является *GSLShell*. Сокращение GSL означает GNU Scientific Library, подборку математических алгоритмов, относящихся практически ко всему потребному при работе с числами: дифференциальным уравнениям, наименьшим квадратам, методам Монте-Карло, преобразованиям Фурье, матрицам...

GSL спасет вас, если надо написать приложение для анализа конечных элементов диетических сухариков, но чтобы просто порезвиться с цифрами и сделать нечто более актуальное, лучшим ответом будет

GSLShell. Как и с любой подобной технической штукой, вам понадобится время на ознакомление с синтаксисом и наличными функциями (знакомство с GSL дает немалое преимущество, поскольку принципы их применения схожи). Но уже довольно быстро вы сумеете нарисовать график, примерно так:

```
import 'math'
require 'plot3d'
f = |x, y| sin(x)*exp(-x^2-y^2)
graph.plot3d(f, -3, -3, 3, 3, {title= '3d plot'})
```

Следует импортировать путь к нужной библиотеке (*GSLShell* написана на Lua, и если вы в нем разбираетесь, это вам пригодится). Затем определяйте функцию и стройте график. Здесь функция $f(x,y)$ яв-



► Быстрая и простая функция построения графиков — только одна из причин, по которым стоит поиграть с GSL.

ляет собой простую математическую поверхность. Вызов `graph.plot3d` принимает в качестве аргументов функцию, затем область ее определения и некоторые дополнительные ключевые слова для задания таких вещей, как заглавие. *GSLShell* использует SDL для рендеринга графика в новом окне. Если вы присвоите построение графика переменной, вы все равно сможете делать дополнения к нему.

«Вам придется потратить время на ознакомление с функциями.»

Клиент IRC

Smuxi

Версия 0.8.9 Сайт <http://www.smuxi.org/main>

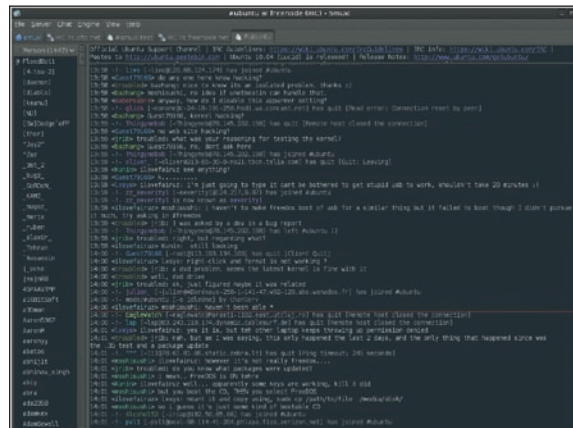
В наши дни придумано немало сложных и разнообразных способов коммуникации, но стоит извлечь урок из того факта, что IRC все еще существует, не говоря уж о том, что для него продолжают создавать программы.

IRC – предтеча Всемирной паутины; этот до смешного простой протокол в основном состоит из написания строковой переменной и легкость реализации делают IRC таким полезным и удобным, но истинная притягательность заключается в культуре пользователей, которые избегают выкрутасов, всячески охраняя изначальную простоту.

Но – ближе к теме. *Smuxi* долгое время был отличным клиентом для IRC, и хотя крутые энтузиасты, признающие лишь командную строку, смотрят на такое свисока, в нем есть весьма неплохие графические функции и в общем приятный и красочный интерфейс. И даже самые закоренелые приверженцы голого текста оценят горячие клавиши, верно?

Однако эта новая версия отчасти впала в ересь, включив поддержку соединения для *Jabber* (это XMPP, или Google Talk, или чат Facebook для вас, в зависимости от ваших предпочтений/лояльности/глупости). Она пока что недалеко ушла от стадии альфа – например, нет возможности управлять списком друзей, а значит, вы будете вынуждены сначала рассылать «приглашения» на соединение через свою обычную клиентскую программу, что, естественно, затрудняет начало общения. Но если разработчики в конце концов умудрятся свести *Jabber* и IRC в одну посуду, *Smuxi* станет весьма полезным приобретением.

В любом случае, это эффективный и приятный способ облегчить себе путь в порой



► IRC IRC — одна из простейших форм доставания всех онлайн. Горды троллингом с 1988!

«Это эффективный и приятный способ облегчить путь в мир IRC.»

запутанный, но по большей части все же дружественный мир IRC.

Компиляция из исходника довольно проста, но требует наличия библиотек Nini и log4net – мы включили исходник для них в состав DVD, чтобы упростить задачу, но вы можете просто найти нужные пакеты в репозитории вашего дистрибутива.

Финансовая программа

moneyGuru

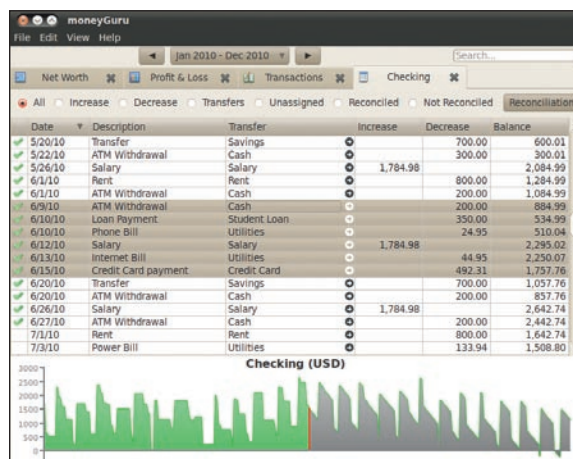
Версия 2.50 Сайт <http://bit.ly/5jua9W>

Планировать свои персональные финансы не более интересно, чем слушать обсуждение бюджета. Проблема большинства финансовых программ в том, что они требуют чересчур больших усилий. Планирование вашего финансового будущего во многом похоже на резервное копирование: вы, разумеется, помните о необходимости им заняться, но вечно отвлекаетесь на нечто повеселее – например, на расставление в алфавитном порядке коллекции DVD или подсчет серебристых машин, проезжающих мимо вашего дома за час.

MoneyGuru стремится отмежеваться от этого правила, став более прямой и оперативным. Настраивать разные счета вам все равно придется, но стало намного проще переключаться между ними и понимать, что там происходит. *Money-Guru* также умеет импортировать данные в стандартных форматах (например, CSV или QIF, столь

любимых большинством онлайн-банков); затем он проанализирует и идентифицирует их для вас – на полное обновление уйдут какие-то минуты. Что еще важнее, *moneyGuru* способен заглядывать в будущее. Получить свой баланс-отчет и узнать, сколько у вас денег, легче легкого, но прежде чем вы спустите их на 2000 цветных ЖК-мониторов из Китая, *moneyGuru*, возможно, спасет вас, указав, что завтра списываются со счета ваши регулярные выплаты по ипотеке/займу/взятке/подписке на *LXF* – это поможет вам планировать бюджет с учетом будущего.

В целом, он умудряется вместить ряд функций настоящей программы для бизнеса (например, работу с различными ва-



► Гребите денюжки обеими руками — и в кубышку их! Новая машина, икра, четырехзвездный отдых...

людями и отслеживание расходов) в приложение, которое выглядит доступным для освоения любым идиотом буквально за пару дней. Мы не собираемся обманывать вас обещаниями, что *moneyGuru* решит ваши проблемы с бюджетом без малейшего усилия с вашей стороны, но он явно облегчит эту задачу, и есть шансы, что вам разрешат остаться в Европейском Союзе.

На главном сайте имеются пакеты для Debian/Ubuntu, хотя компиляция из исходника не особо трудна.

«MoneyGuru стремится отмежеваться, став более оперативным.»

Программа визуализации сети

IP-Link

Версия 0.2 Сайт <http://ip-link.wikidot.com>

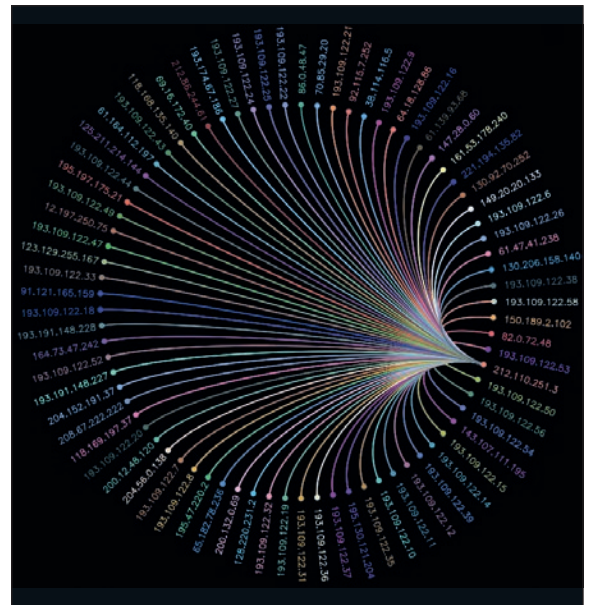
Всем нравится красивая графика, особенно та, в которой много данных. Иногда они бывают даже полезны, но главное – они красивы. Масса сетевых инструментов пытаются создать какие-либо диаграммы на основе найденного, но вам будет приятно узнать, что *IP-Link* – не из таковых. Ну, сам по себе.

Это даже не одно приложение, а скорее набор скриптов Python, работающих с вашими данными. Да, он даже не озабочится снимком данных сети – вам придется делать его самим. К счастью, командная утилита *tcpdump* берет это на себя. Воспользуйтесь *tcpdump* (если вы этого не умеете, то неплохое руководство есть на <http://linux.byexamples.com/archives/283/simple-usage-of-cpdump/>), чтобы сохранить файл *pcap*. Заполучив этот файл, скрипты *IP-Link* примутся за работу. Вы можете перевести файл в базу данных *SQLite* (удобно для аналитики), или, что более эстетично, в последовательный объект – его затем можно обработать всякими инструмента-

ми, которые не все одинаково «полезны», но все достаточно красивы.

Два самых интересных способа визуального отображения такого вида информации – это *moowheel*, где все адреса рассажены по кругу, а соединения показаны в виде линий, и *circos*, с похожим круговым отображением соединений, но с куда большим запасом опций по части отображения взаимоотношений (число соединений, трафик и все прочее). Создание графика *circos* требует ряда действий и соответствующей программы (которую вы без проблем установите из CPAN). В *circos* можно завязнуть надолго, но, пошарив на сайте *IP-Link*, вы найдете там готовую конфигурацию.

На домашней странице вполне достаточно документации по использованию



» *Moowheel* – это не инновационный инструмент для отгона рогаатого скота, а весьма неплохой способ отображения информации.

«Справляется с данными из интенсивно вкалывающей сети.»

этой программы, и прилагается несколько видеороликов по созданию различных визуализаций. Попробуйте! Лучше всего оно справляется с данными из интенсивно вкалывающей сети.

Генератор случайных чисел

Haveged

Версия 1.4 Сайт <http://www.issihosts.com/haveged>

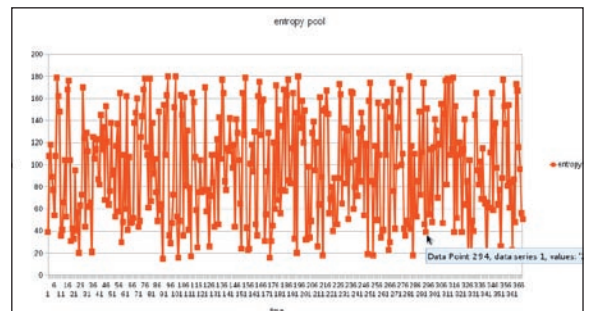
Создание случайных чисел – слишком важная задача, чтобы отдать ее на волю случая. Возможно, это звучит, как рекламный слоган от *LXF*, но создал его участник ядерного проекта «Манхэттен», физик, шахматист и эксперт в области псевдослучайных чисел Роберт Ковью [Robert R. Coveyou]. Случайные числа крутятся во множестве областей, самая известная из которых – системы безопасности. Если вы можете найти следующее число в якобы случайной последовательности, тем легче вам взламывать коды и заниматься самыми разными правонарушениями.

В Linux есть отлично продуманный генератор псевдослучайных чисел – он живет в *dev/random*, и если вы попросите у него число, он вам его выдаст. Однако код, который генерирует эти числа, полагается на энтропию системы – присущую компьютеру случайность всего происходящего в данный момент времени: задержки

между нажатиями на клавиши, движения мыши и т.п. – все вносит свой вклад. Однако на сервере без клавиатуры, мыши и монитора этот вид энтропии отсутствует, и генерируемые числа становятся предсказуемы – а это опасно.

Возможно, для бытового применения они по-прежнему будут неплохи, но если требуется обеспечить подлинную случайность ваших чисел, стоит обратить внимание на *haveged*. Это не генератор чисел – он просто использует разные методы «сбора» энтропии с системы, и эти собранные данные используются обычными алгоритмами. *Haveged* вносит дополнительную энтропию, запуская детерминированные процедуры и затем отмечая различия,

«Вносит энтропию, запуская детерминированные процедуры.»



» Хмм. Случайные числа не очень-то изобразишь. А это график изменения энтропии системы со временем...

например, во времени запуска сложного процесса – даже на 100 % холостом процессоре разные вещи будут происходить по-разному, в зависимости от состояния кэша, процедур прерывания и еще миллиарда и одной причины, проистекающих из современных CPU.

В компиляции и установке *haveged* – сама простота. Однако если вы хотите использовать его для важных задач, было бы разумно сначала проверить, будет ли он работать так, как нужно для вас. Простейший тест на случайность данных – заполнить ими большой файл и попытаться сжать его; все повторяющиеся фрагменты будут оптимизированы, и если в итоге вы получите файл, намного меньший по размеру, значит, ваши числа не совсем случайны.

HotGames Развлекательные приложения

Артиллерийская/танковая игра

Scorched 3D

Версия 43.3 Сайт <http://www.scorched3d.co.uk>

Похоже, эта игра эксплуатирует такую идею, что взрывание чего-нибудь при посредстве коварства доставляет нам удовольствие... Особенно если это «что-нибудь» – несчастные игроки. Идея плодотворная; и если кто-нибудь за нее зацепится...

Главная потеха в игре начинается, когда вы присоединитесь к одному из популярных серверов и попытаетесь уничтожить реальных соперников на множестве различных территорий. Не имея ничего, кроме законов Ньютона и огромного реестра примочек себе в помощь, вы быстро выучитесь думать о скорости и стратегии.

В конце каждого раунда вы получаете призовые деньги, сумма которых зависит от нанесенного вами урона. К игре придется привыкать – это вам не 2D-мир, которого вы, вероятно, навиделись в других играх данного типа. Чтобы по-

мочь вам прицелиться, перед выстрелом появляется красный треугольник. Он настроен в соответствии с текущей позицией вашей орудийной башни – прицельтесь, представив, что ваш снаряд полетит в направлении верхнего угла треугольника (но сила ветра и гравитация ему мешают). Возможно, чтобы освоиться, вам придется предпринять несколько попыток, но как только это произойдет, вам наверняка захочется дорваться до всех карт и протестировать свои новые разрушительные навыки онлайн.

Если вы хотите скомпилировать игру из ядра, нужно сделать несколько особых шагов. Вначале запустите скрипт Perl:

«Вам наверняка захочется протестировать свои навыки онлайн.»



› Разрушайте все. А потом выходите в режим онлайн и разрушайте других людей и их собственность.

```
cd scorched/scripts
perl createAMMmakefile.pl
cd ..
sh ./autogen.sh
make
sudo make install
```

Предупреждаем, что компиляция отнимает много времени – минут 30 и более. Гораздо проще просто взять один из готовых бинарников – они заготовлены почти для всех дистрибутивов.

Стрелялка

Bitfighter

Version 016 Сайт <http://bitfighter.org>

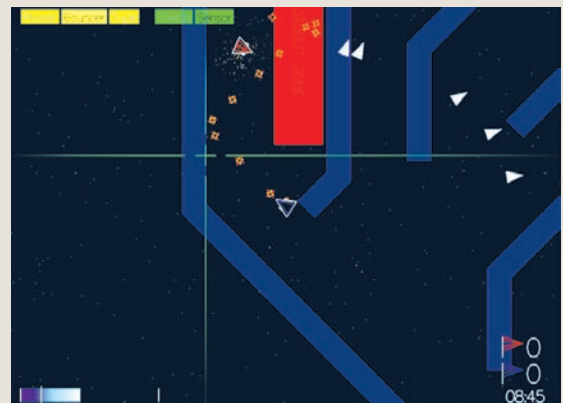
Вы когда-нибудь пытались представить себе семимерное пространство? О нет, не пытайтесь! У вас пойдет кровь носом. Короче, чем больше у вас пространств, тем сложнее все становится, и вот поэтому-то *bitfighter* придерживается старой школы тех дней, когда у вас могло быть сколько угодно измерений, лишь бы на самом деле их оставалось два (справедливости ради отметим, что одномерные игры никогда не имели настоящего успеха – кое-кто считал их непомерно линейными, в чем, вероятно, была доля истины). Примерно такое получилось бы, если бы у *Asteroids* было чуть больше памяти, цветной графики, игровых режимов и добавочных игроков.

Стреляйте, лавируйте, маневрируйте и уворачивайтесь на разных уровнях на своем треугольном космическом ко-

рабле и принимайте участие в захвате флага, футболе, *Nexus* (это как захват флага, только не в команде) и в нескольких других режимах.

Корабль можно обновлять и оснащать добавочными модулями и вооружением, и выбор препятствий, с которыми вы будете сталкиваться, достаточно богат. Модули придают кораблю свойства вроде улучшенной брони, так что вам придется принимать тактические решения, прежде чем палить. Одно из новаторских дополнений – «инженерный» модуль, позволяющий игроку собирать собственные орудийные башни или силовые поля.

«Когда вы прочувствуете инерцию, игра пойдет органично.»



› *Bitfighter* – вот что случилось бы, появившись Всемирная паутина до изобретения видеокарт.

Когда вы прочувствуете инерцию, игра пойдет вполне органично. Во многих аспектах, упрощенная графика и векторные игровые области очень помогают – было бы слишком сложно управлять кораблем и одновременно следить, что происходит, в визуально перенасыщенной обстановке.

Да, в *bitfighter* действительно следует играть онлайн, но серверов и игроков в сети хватает на битву сутками напролет. И да, сначала вы хлебнете горечи поражений, но не сдавайтесь – упорство вознаграждается.

Моделирование толпы

PedSim

Версия 2.2 Сайт <http://pedsim.silmaril.org>

Отвлекаться на выяснение не стоит трудов, но уж хоть какие-нибудь исследования да пришли к выводу, что толпа представляет собой некую разновидность мета-организма. Почти как человек, который к концу дня становится полуроботом и самодостаточной колонией бактерий.

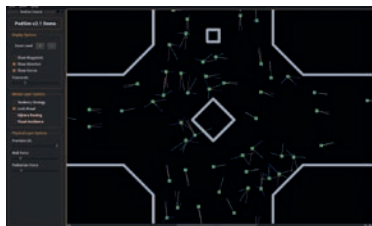
Однако работа с микро- и макроскопической природой толпы важна – и хотите ли вы узнать, сколько времени уйдет на эвакуацию здания, или вам просто любопытно взглянуть, что собой представляет огромная очередь в центре занятости, смело обращайтесь к *PedSim*.

Помимо хорошо документированной библиотеки, здесь есть демо-приложение, объясненное ничуть не хуже и способное работать с простыми имитациями. Входящие данные определяются как файл ввода XML, где перечислены заграждения и промежуточные точки траектории движения, а также люди. Есть способ определять группы людей и даже выделять их в пространстве. Вот простой пример:

```
<agent x="60" y="0" n="100" dx="70"
dy="10">
<addwaypoint id="w1" />
<addwaypoint id="w2" />
</agent>
```

Здесь имеется 100 субъектов, сгруппированных вокруг точки 60,0. Величины dx и dy определяют область, где они будут рассредоточены, и даны промежуточные точки маршрута, куда они стремятся попасть. Также добавлены заграждения.

Демо-приложение прекрасно отображает действия, но в крайнем случае вы можете экспортировать данные и использовать их в системе рендеринга (например, POV-Ray).



➤ Нет, это не устройство слежения инопланетян. Эти точки — люди и места, куда они направляются.

Программа для просмотра документов

Evince

Версия 3.35 Сайт <http://projects.gnome.org/evince>

Документы полезны... это если их можно прочесть. *Evince* гордо объявляет себя приложением-прибежищем для прочтения любого вида файлов на рабочем столе Gnome. Его слоган – «Просто просмотрщик документов», но задача сделать что-то столь фундаментальное так удачно отнюдь не проста.

И дело не в том, что он поддерживает самые разные типы файлов и умеет работать с самыми важными (PDF, PostScript, DejaVu и всякие форматы «комиксов» типа CBZ); дело в том, что он справляется с этим блестяще, и даже постоянный поток усовершенствований пока что не смог его испортить. В последние релизы добавлены более удобные комбинации клавиш и новая кнопка Send To, которая использует Dbus для ускоренной обработки ваших документов. Восторги по поводу просмотрщика документов могут показаться придуховатыми, но *Evince* – действительно

отличная программа, и одна из немногих реализаций просмотра PDF, работающих, как положено, включая распечатку.

Если вы пользователь Gnome и постоянно следите за обновлениями, ждите в скором времени появления новых версий. А если вы не имеете привычки именно к этому рабочему столу, *Evince* стоит установить. Скомпилировать его из имеющегося исходника нетрудно, но, возможно, лучше дождаться появления свежего пакета в репозитории вашего дистрибутива. LXF



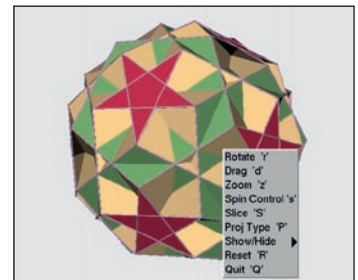
➤ Ну да, это всего лишь просмотрщик информации, но он так хорош, что заслуживает любви.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

➤ Antiprism 0.2.0

Моделирование многогранников во всей геометрической красе. И к тому же с цветным отображением! <http://www.antiprism.com>



➤ Искрометную шутку про додекаэдр вставляйте сюда сами.

➤ tor 0.2.3.12-alpha

Для тех, кто озабочен безопасностью, лучшая защита снова улучшена. <https://www.torproject.org>

➤ PHPMyAdmin 3.4.10

Зачем пачкать руки об базы данных, если всю грязную работу выполняет симпатичный сетевой интерфейс? http://www.phpmyadmin.net/home_page/index.php

➤ pf-kernel 3.2.5

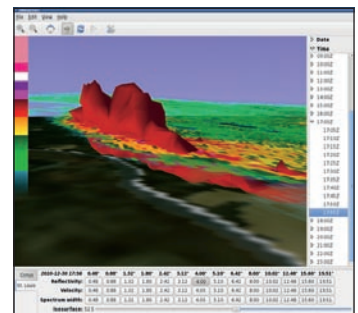
Слегка модифицированная версия простого релиза ядра, с дополнительными удовольствиями наподобие гибернации TuxOnIce. <http://pf.natalenko.name>

➤ VMS-Empire 1.9

Оригинальная имперская игра (как *Civ*, но с консольной графикой), переработанная под современные компиляторы. <http://www.catb.org/~esr/vms-empire/>

➤ Aweather 0.7

Еще одно недавно упомянутое приложение: отслеживание климата включило поддержку GPS. <http://lug.rose-hulman.edu/proj/awweather>



➤ Метеослужба Англии освободит свои данные этак после дождичка в четверг.

На диске

Опробуйте наши среды рабочего стола



В этом месяце мы демонстрируем несколько новых сред рабочего стола. С виду герои дня — Unity и Gnome 3, но они вовсе не обязательно будут самым лучшим выбором. Как известно всем умникам, удобство использования рабочего стола — в глазах смотрящего. Мы рассказываем о дебютанте состязания — слушатели подкаста уже знают фаворита автора

(Cinnamon); об одном из испытанных столбов (KDE); и о малютке, становящемся объектом культа (OpenBox). И у каждого — своя интерпретация парадигмы рабочего стола; решайте, которая из них подходит лично вам. Мы также предлагаем вам самые свежие релизы дистрибутивов — Sabayon (с Gnome), и, конечно, код из руководств и множество программ из журнала. Наслаждайтесь!

Передовой и простой в использовании

Sabayon

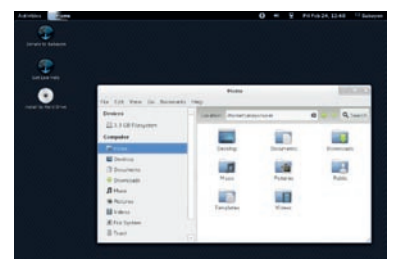
Sabayon основан на Gentoo, но пусть вас это не пугает. Хотя его родитель — огнедышащий дистрибутив для суперпрофессионалов, где все компилируется из исходника, Sabayon стремится к тому, чтобы обеспечить передовые, но удобные в использовании условия работы. Вам даже не придется ничего компилировать.

Одна из причин столь сильного отличия родителя от своего дитя — менеджер пакетов. Gentoo построен вокруг Portage, который загружает и устанавли-

вает исходный код, а Sabayon применяет Entropy, который, как и большинство других менеджеров пакетов, использует прекомпилированные бинарники. Если для вас это пустой звук, не переживайте. Вам нужно только знать, что для установки пакета надо перейти в Entropy Store и найти приложение точно так же, как вы бы сделали в любом другом дистрибутиве.

Из-за способа Sabayon сканировать live DVD нам пришлось включить два пункта в меню загрузки в зависимости от того, загружаетесь ли вы с DVD или перенесли его на устройство USB. Загрузка с USB включает пункт, указывающий на устройство, где находится дистрибутив. По умолчанию мы установили его в `/dev/sdb1`, но если у вас больше одного жесткого диска, вам, возможно, придется это изменить. В меню

Sabayon USB стрелками выберите опцию, с которой вы хотите загружаться. Нажмите на F5, затем на Esc, чтобы активировать редактирование, и измените `cdroot=/dev/sdb1`, указав на нужное устройство. После этого, нажав на Enter, чтобы начать загрузку.



► Sabayon: не просто итальянская закуска.

«Sabayon стремится обеспечить передовые, но удобные условия.»

Старое встречается с новым

OpenSUSE

Первое, что вы увидите при запуске этого дистрибутива — улыбающегося хамелеона по имени Гико [Geeko]. Второе — это классная новая среда рабочего стола, разработанная Клемом Лефевром [Clem Lefebvre] и парнями из Linux Mint. Мы сделали ремикс 2012 года для старейшего коммерческого дистрибутива, а благодаря чудесам SuseStudio сверху присыпали Cinnamon [Cinnamon — англ. корица].

Если вы использовали какую-либо среду рабочего стола, отличную от Gnome Shell или Unity, последние лет десять, в Cinnamon вы сразу же почувствуете себя как дома. Для внесения корректировок в рабочий стол перейдите в Menu > Preferences

> Cinnamon Settings. Нам нравится тема Dark-Glass, но это наш личный выбор. Некоторые изменения не вступят в силу до тех пор, пока вы не выйдете из системы и не вернетесь назад. (Используя имя пользователя 'tux' и пароль 'linux').

Испытайте его, а если вы решите, что Cinnamon — лучший пользовательский интерфейс, то его пакеты доступны для большинства основных дистрибутивов на cinnamon.linuxmint.com, если вы не желаете устанавливать OpenSUSE.

Если для запуска Cinnamon у вас не хватает графических возможностей (например, вы используете виртуальную машину), то он загрузится в Gnome Shell. Из-за процесса загрузки, использован-



ного SuseStudio, он не запустится, если образ DVD перенесен на флэшку. Как бы то ни было, образ жесткого диска, который может быть перенесен на USB-устройство, доступен на susestudio.com/a/jMOVxa/opensuse-cinnamon (у нас на DVD не хватило места на него).

► Хамелеон со вкусом корицы. М-ммм.

Рабочий стол без всяких этаких глупостей

CrunchBang

Одни любят большие среды рабочего стола, другие считают их громоздкими. Для этих других мы включили CrunchBang. В отличие от большинства малых дистрибутивов (вроде тех, что были на диске прошлого месяца), CrunchBang предназначен не для старых машин, а для того, чтобы работать быстро, не тратя времени на ерунду. Как раз поэтому он и на старых машинах тоже работает хорошо. Для достижения данной цели он выбрал за основу OpenBox, минималистский менеджер окон.

OpenBox существует уже не первый день, но пока особо не преуспел – в основном потому, что не особо пытался. Это рабочий стол не для каждого, а для элиты, которая понимает, что меньше – значит лучше. При знакомстве с ним сразу бросится в глаза отсутствие любого вида меню на панели – это сводит к минимуму все помехи на рабочем столе. Но не бойтесь, он не по-

шел по пути Gnome и Unity и не ликвидировал меню как класс: стоит щелкнуть правой кнопкой по рабочему столу, как оно появится. Для более сложных задач на рабочем столе имеется удобный список ярлыков. OpenBox разработан с учетом возможностей настройки, и если что-то в нем вам не нравится, вы наверняка сможете это изменить. OpenBox имеется в большинстве дистрибутивов, так что вам не обязательно устанавливать ради него CrunchBang.



CrunchBang возвращает вас к основам.

KDE во всей элегантности

Chakra

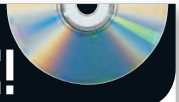
Мы недавно рассказывали вам о Chakra, но снова к ней вернулись, потому что новый релиз дает вам возможность познакомиться с KDE 4.8 до того, как большинство дистрибутивов обновят свои ISO. На этот раз мы решили включить еще и 32-битную версию, чтобы те из вас, кто в прошлый раз лишился удовольствия из-за более старой версии оборудования, смогли попробовать ее теперь.

Будущее этой среды рабочего стола вызвала весьма оживленные дебаты, поскольку Canonical прекратили финансирование Kubuntu, но факт остается фактом – она предлагает пользователям привлекательный внешне, интуитивный рабочий стол в традиционном стиле.

Эта последняя реализация уже заслужила пылкую любовь тех, кто обывка в этой среде. Похоже, это самая крупная среда рабочего стола из всех существующих, и с максимумом функций, из-за чего она вызывает равно и похвалу, и критику.

Если вы – новичок в KDE, загляните в Activities. Это способ организовать ваш рабочий стол в соответствии с родом ваших занятий. Например, вы, возможно, захотите, чтобы у вас была разная настройка рабочего стола для работы и отдыха. Чтобы начать работу с Activities, щелкните по значку с тремя точками рядом с кнопкой меню в нижнем левом углу (но помните, что все внесенные вами изменения будут потеряны при выключении компьютера, если вы не установили Chakra).

Важно ВНИМАНИЕ!



Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

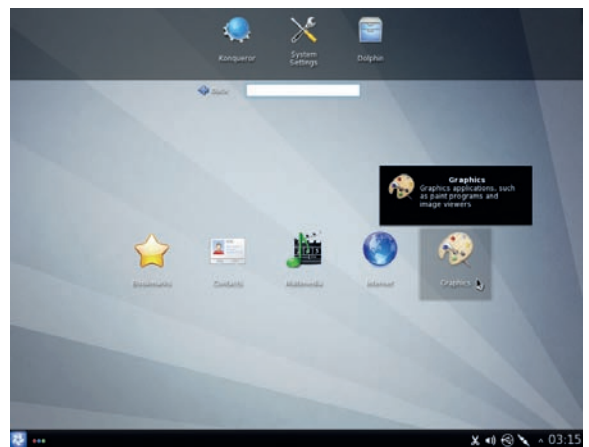
Диски Linux Format DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция Linux Format не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска Linux Format, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru, для получения содействия.



Любитесь, гномофилы, насколько хорошим мог бы быть ваш рабочий стол.

«Самая крупная среда рабочего стола, с максимумом функций.»

На диске

Помимо четырех симпатичных дистрибутивов, мы включили в состав диска весь код из руководств, все программы из HotPicks и все программы, упомянутые в журнале, рядом с которыми вы видите знак «На диске». Если вы установите DVD в компьютер, он должен открыть страницу HTML по умолчанию, но если ваши настройки безопасности не позволяют этого, перейдите в [index.html](#), чтобы открыть для себя все 4.4 ГБ прелестей Linux.

Для ознакомления с дистрибутивом, установите диск в дисковод и перезагрузите компьютер. Появится экран, где вы сможете выбрать то, что вам нужно. Если ваш компьютер загружается в обычном режиме, а не с диска, вам придется изменить настройки BIOS на загрузку с DVD. Если вам захочется перенести содержимое диска на устройство USB, мы включили для вас удобный скрипт – как им пользоваться, описано в [dvd2usb.html](#) на диске.

Цифровые подписчики, обратите внимание, что вы можете просто использовать `dd` для перемещения загруженного ISO на устройство USB. Используйте `sudo dd if=lx157.iso of=/dev/sdX` при необходимости включая путь к загруженному ISO, где X изменяется соответствующим образом для устройства USB. ЭТО УНИЧТОЖАЕТ ВСЕ ДАННЫЕ НА УСТРОЙСТВЕ. ЗАДАВ НЕ ТОТ ДИСК, ВЫ МОЖЕТЕ СТЕРЕТЬ ВСЕ ДАННЫЕ НА ЖЕСТКОМ ДИСКЕ.

Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам. Но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в «Линуксцентр» по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:

 <p>LXF154 Февраль 2012</p> <ul style="list-style-type: none">» Ноутбук с Linux Как его выбрать» Узрим звезды Астрономические программы» BrowserID Идентификация по данным электронной почты» Скрипт мыши Автоматизируем щелчки и перемещения курсора <p>LXFDVD: Linux Mint 12, BackTrack Linux 5 R1, ядро 3.0, CRUX 2.7.1, Shutter и многое другое...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_154/ PDF-версия: shop.linuxformat.ru/elxf_154/</p>	 <p>LXF155 Март 2012</p> <ul style="list-style-type: none">» Открытие Linux Как превратиться из салаги в гуру» Менеджерам на заметку Планировщики проектов» Dart Язык программирования от Google» Клевый музон Коллекция винтажных синтезаторов <p>LXFDVD: Ubuntu 11.10 LXF Remix и еще 6 загружаемых дистрибутивов, CloneZilla, Condor Cloud, SELinux, игры и многое другое...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_155/ PDF-версия: shop.linuxformat.ru/elxf_155/</p>	 <p>LXF156 Апрель 2012</p> <ul style="list-style-type: none">» Raspberry Pi Компьютер-крошка делает революцию» Не стареют душой ветераны Продлеваем жизнь компьютерам» Dracut Инструмент ускорения загрузки, не зависящий от дистрибутива» Видеонаблюдение Сам себе Большой Брат <p>LXFDVD: Linux Gamers Live и еще 6 загружаемых дистрибутивов, ZoneMinder, FontForge, FileMonitor и многое другое...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_156/ PDF-версия: shop.linuxformat.ru/elxf_156/</p>
---	--	---

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала — оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru или shop.linuxformat.ru получают электронную версию в подарок!

Подписывайтесь на журнал на www.linuxformat.ru/subscribe/

Телефоны отдела подписки: Санкт-Петербург (812) 309-06-86, Москва (499) 271-49-54

Специальное предложение

Купите подборку журнала!

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски LXF. Если вы тоже этим озадачены, то в интернет-магазине «ГНУ/Линуксцентра» продолжается продажа журналов за 2010 год. Вы можете приобрести как отдельные номера изданий, так и подписки на 6 или 12 месяцев. Спешите — журналов осталось не так уж много!

shop.linuxformat.ru



6 месяцев
900 руб.

12 месяцев
1800 руб.

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 по списку: «tgz» – это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!



CentOS 6.2 Sabayon 8

» Ядро 2.6.32

» Gnome 2.28 и KDE 4.3.4

» OpenJ/KVM

» SELinux

64-разрядная сборка

И еще 3 загружаемых дистрибутива

- » Шкара 2012.2
- » Smpcbang 10
- » openSUSE 12.1 Smpcbang Remix

А ТАКЖЕ: Oracle/AS, средства обеспечения безопасности, полная совместимость с RHEL

А ТАКЖЕ: Dnsmasq, IP-Link, RedSim и многое другое...



Май 2012
LXF DVD 157

LINUX
FORMAT

Май 2012
LXF DVD 157

LINUX
FORMAT

ДИСТРИБУТИВЫ

- CentOS 6.2** Установочный DVD. 64-разрядная сборка (вторая сторона **LXF DVD**)
- Chaika 2012.2** Live-CD с наиболее свежей версией KDE 4.8 (загрузка с **LXF DVD**)
- CunchBang 10** Дистрибутив с оконным менеджером OpenBox (загрузка с **LXF DVD**)
- openSUSE 12.1** Sinnamon Remix Сборка популярного дистрибутива с Gnome 3.2 и Sinnamon (загрузка с **LXF DVD**)
- remastersys 3.0.0** Средство создания Live CD для дистрибутивов на основе Ubuntu

НОТІСЬКІСЬ

- WiFiFighter 0.16** Игра, космическая война в ретро-стиле
- DiffImage 1.2.1** Графическая утилита сравнения изображений
- Evince 3.3.5** Программа просмотра документов (PDF, djvu, Postscript и др.) для Gnome
- GSLShell 2.1.0** Интерактивный командный интерфейс к GNU Scientific Library
- Navedit 1.4** Улучшенный по сравнению с /dev/random генератор случайных чисел
- IP-Link** Программа визуализации сетевого трафика, написанная на Python

MoneyGuru Программа для управления

- персональными финансами
- Panther 0.9** Простой конвертор форматов аудио- и видеофайлов
- PedSim 2.2** Средство моделирования поведения больших групп пешеходов
- Scorched3D 4.3.3** Игра, 3D-стрелялка по мотивам Scorched Earth
- Smuxi 0.5.5** Дружелюбный пользовательский интерфейс для IRC

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@lxfmagazine.com

Дорожные диски В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу disk@lxfmagazine.com

Настоящий диск тщательно тестировался и проверялся на всех старых платформах, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, с сайта www.linuxformat.ru скачайте и установите все файлы данного приложения.

Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАО 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего 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
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под 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?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: 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.



ФОРУМ №1

ДЛЯ ВСЕХ
ПОЛЬЗОВАТЕЛЕЙ
LINUX



LINUXFORUM.RU

Linux center
www.linuxcenter.ru

Отдел
дистрибуции
ГНУ/Линуксцентра
приглашает
дилеров
и дистрибьюторов
к сотрудничеству!

Широкая сеть
представительств
в разных городах
позволит вам
оптимизировать
процессы логистики
и доставки товара.

Подробнее о партнерской программе:
www.linuxcenter.ru/partner/

FLEXTRON Axis –

рабочая станция для компьютерного моделирования и CAD-приложений на основе профессионального графического процессора NVIDIA Quadro 2000



РЕКЛАМА

Освободите время для творчества!

Революционная архитектура NVIDIA® Fermi объединяет продвинутую визуализацию и вычислительные возможности, обеспечивая производительность, многократно ускоряющую привычный рабочий процесс.



Основные характеристики:

Процессор Intel Core i7-2600
Графический процессор NVIDIA Quadro 2000, 1024MB
Платформа ASUS P8P67 Pro
Оперативная память 16GB
SSD 90GB
Жесткий диск 2000GB
Корпус IN-WIN BUC666 600BT

ЦЕНА ОТ
51 000
РУБ



www.fccenter.ru www.fcshop.ru

LINUX FORMAT

Главное в мире Linux

Как разместить рекламу в разделе Classifieds?

¼ полоса (210 × 297 мм)	165 200 руб.
½ полосы горизонтально (197 × 144 мм)	88 500 руб.
½ полосы вертикально (102 × 278)	88 500 руб.
¼ полосы вертикально (98 × 138 мм)	53 100 руб.
Фотоблок (44 × 113 мм)	15 000 руб.

Тел.: +7 812 309 06 86

Цены указаны с учетом НДС



Возможности в надежности Ноутбук iRU Patriot 510 15,6"



На базе процессора Intel® Core™ i3
второго поколения



- Стильный внешний вид
- Компактный размер
- Низкое энергопотребление
- Идеальный мультимедиа центр



Intel, логотип Intel, Intel Inside, Intel Core и Core Inside являются товарными знаками корпорации Intel на территории США и других стран.

ГНУ/Линуксцентр приглашает на работу!



ВАКАНСИЯ: Drupal-разработчик

ОБЯЗАННОСТИ:

- » Создание сайтов с нуля на базе CMS Drupal: верстка, программирование, разработка баз данных, конвертация данных.
- » Разработка дополнительных модулей для CMS Drupal.
- » Внесение изменений в готовые скрипты.
- » Поддержка сайтов компании.

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

ГНУ/Линуксцентр приглашает на работу!



ВАКАНСИЯ: Контент-специалист (младший веб-мастер) в отдел маркетинга

ОБЯЗАННОСТИ:

- » Поддержка в актуальном состоянии контента на сайтах компании.
- » Работа с контентом, HTML- и PHP-кодом.
- » Работа с каталогом товаров интернет-магазинов, размещение новых товаров.
- » Участие в интернет-проектах компании.
- » Составление и отправка новостных рассылок клиентам и партнерам.
- » Сотрудничество с веб-программистами и системными администраторами.
- » Работа с системами контекстной рекламы, баннерообменными сетями.

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

Школа LXF

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Взаимодействуем с окружением, или И роботы чувствовать умеют

Татьяна Казанцева продолжает рассказывать о робототехнике, которая обходится без реальных роботов.



Наш эксперт

Татьяна Казанцева в свободное время от корпения над написанием методики скрещивания Scratch и Arduino оттачивает навыки работы со свободным ПО для использования в школе и дома.

Сегодня речь пойдет о датчиках, которые позволят нашему роботу ориентироваться в пространстве. В прошлом выпуске Школы (LXF156) мы начали разбираться, как можно изучить основы робототехники в условиях, когда реальные роботы недоступны. Мы научили робота двигаться, разобрались, как им можно управлять, и даже учли возможности внесения искажений в передаваемые команды.

Итак, мы научим виртуального робота взаимодействовать с окружением, оснастив его датчиками (конечно, тоже виртуальными), которые будут отражать реально существующие компоненты при построении реальных роботов.

Коснись меня

Самый простой датчик, которым оснащают роботов, это датчик касания (или бампер). Такие датчики позволяют роботу определять наличие препятствий по ходу движения, прикоснувшись к ним. Самый простой датчик – это датчик касания, расположенный по ходу движения робота. Для демонстрации движения нам понадобится также виртуальный лабиринт, по которому наш робот будет разъезжать.

Робот в нашем случае будет представлять собой круглую платформу, спереди которой располагается датчик касания. Сразу

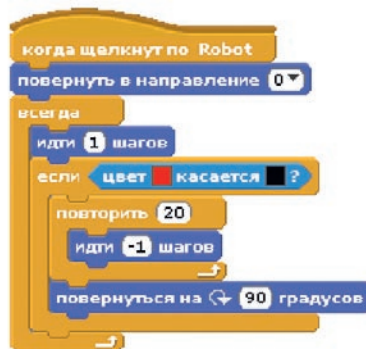
стоит отметить, что рисовать датчики надо на правой стороне робота, так как именно туда смотрит направление движения 0.

Также нам понадобится поле для его запуска, которое в нашем случае будет иметь вид пространства, окруженного сплошным забором.

Алгоритм робота будет следующим:

- 1 Робот едет вперед, пока не коснется забора
- 2 Если срабатывает датчик касания, то робот отъезжает назад и поворачивается на 90 градусов по часовой стрелке.
- 3 Цикл повторяется.

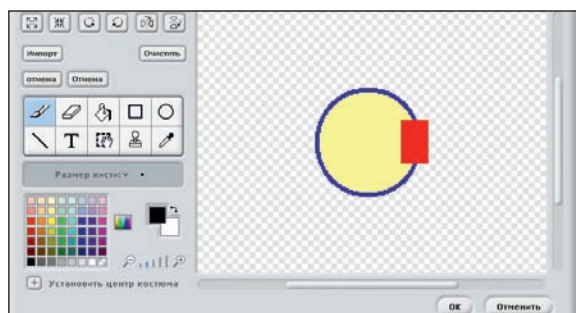
А программа будет такой:



Запускаем мы робота щелчком по нему (это позволяет нам поставить его в любую точку поля).

Как видно, мы используем цвет датчика (красный) для определения соприкосновения с цветом стены (синий). Это позволит нам в дальнейшем использовать несколько датчиков.

Усложним задачу. Используем лабиринт, показанный на рис. 2. Поставив робота в его начало и запустив, вы можете обнаружить, что он застрял в средней его части и не может выбраться. Что же делать? Попробовать изменить поворот в направлении против часовой стрелки? Тогда робот будет постоянно возвращаться назад.



► Рис. 1. Вот так будет выглядеть наш робот.

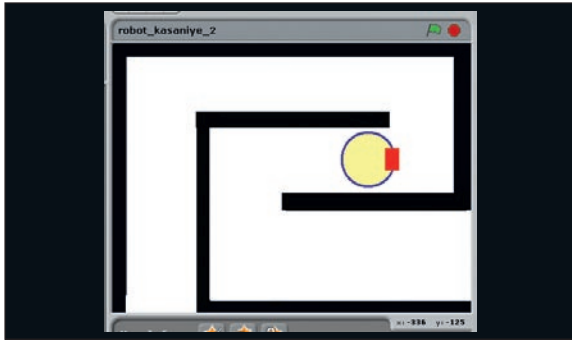
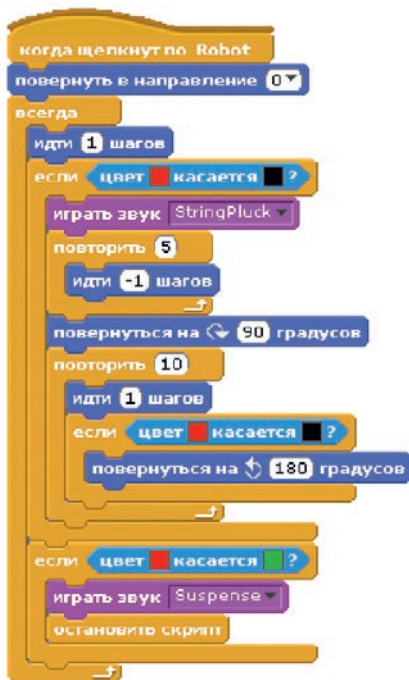


Рис. 2. Заведем нашего робота в лабиринт, и пусть он попробует добраться до упора.

Можно сделать следующее:

- 1 Столкнувшись со стенкой, робот поворачивается вправо и пробует поехать.
 - 2 Если он снова упирается в стенку за минимальное число движения (к примеру, на величину отъезда, умноженную на два), то он поворачивается на 180 градусов.
- Видоизмененный код программы будет выглядеть теперь так.



Мы добавили еще стоп-линию зеленого цвета (нарисуйте ее в конце лабиринта) и звуки при соударении со стенкой и успешном прохождении лабиринта (рис. 3):

Позэкспериментируйте с разными лабиринтами и алгоритмами.

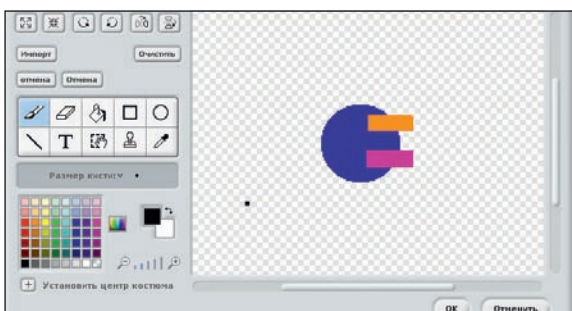


Рис. 4. Новый робот — чуть меньше, зато с двумя датчиками.

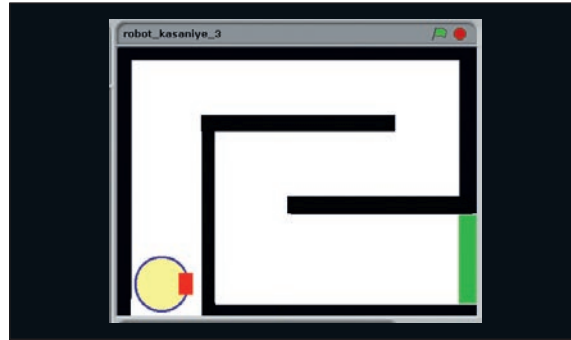


Рис. 3. Наш робот теперь сумеет остановиться при достижении конца лабиринта.

Усложним задачу

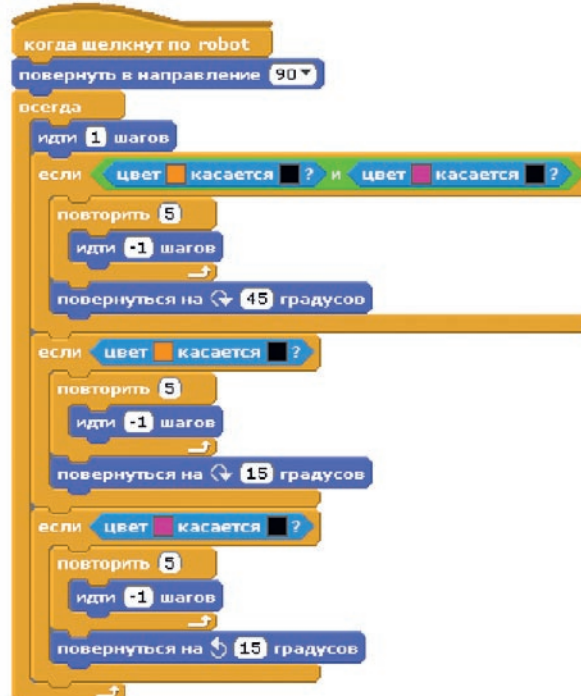
Теперь усложним нашего робота и поставим на него два датчика — справа и слева по движению, и пустим робота ездить по полю с разбросанными кубиками (рис. 4).

Так как у нас теперь два датчика касания, то мы сможем точнее позиционировать робота и определять нахождение препятствий.

Но и алгоритм работы также усложнится:

- 1 Робот движется вперед.
- 2 При срабатывании датчика касания робот отъезжает назад.
- 3 Если сработал правый датчик, то мы поворачиваемся на небольшой угол (10–20 градусов) влево.
- 4 Если сработал левый датчик, то мы поворачиваемся вправо.
- 5 Если сработали оба датчика, то мы поворачиваемся вправо на 45 градусов.

В блоках Scratch это будет выглядеть так:



Запустив программу, вы через некоторое время можете столкнуться с тем, что ваш робот упрется в угол и не сможет из него выйти. Это связано с ограничением алгоритма, который при переменном касании датчиков поворачивается на определенный угол то вправо, то влево.

Выйти из положения можно с помощью увеличения угла поворота при последовательных касаниях или просто задавая угол случайным числом (от 10 до 20 градусов будет достаточно).

Тогда программа примет следующий вид:

»


```

когда щелкнут по robot
  повернуть в направление 90
  всегда
    идти 1 шагов
    если цвет [касается ?] и цвет [касается ?]
      повторить 5
        идти -1 шагов
        повернуться на 45 градусов
    если цвет [касается ?]
      повторить 5
        идти -1 шагов
        повернуться на [выдать случайное от 10 до 20] градусов
    если цвет [касается ?]
      повторить 5
        идти -1 шагов
        повернуться на [выдать случайное от 10 до 20] градусов
  
```

Да будет свет

Следующий датчик, который мы рассмотрим и попытаемся смоделировать, будет датчиком освещения. В реальности это обычно фоторезистор или фототранзистор, который, реагируя на уровень света, меняет свою характеристику (к примеру, сопротивление). За счет преобразования в Scratch будет передан сигнал, который будет изменяться от 0 до 100, где 0 – минимальный уровень освещения, 100 – максимальный.

Источник света у нас будет реализован в виде спрайта конуса освещенности, который можно будет передвигать. Работа может быть взята из первого опыта, разместив источник света посередине спереди.

Итоговое поле будет выглядеть, как показано на рис. 5.

Алгоритм работы будет такой:

- 1 После включения робот начинает крутиться на месте, ища источник света.
- 2 Если он его находит, то начинает ползти к нему.
- 3 Как только уровень сигнала падает, робот опять начинает крутиться на месте.

А программа получится такая:

```

когда щелкнут по Robot
  всегда
    если цвет [касается ?]
      идти 1 шагов
    или
      повернуться на 15 градусов
  
```

Вы можете самостоятельно доработать программу и использовать два датчика света, как и в случае датчиков касания. Также вы можете ввести переменную, чтобы смоделировать изменение уровня освещенности при перемещении от источника или к нему. Это позволит роботу двигаться в правильном направлении, так как в текущем алгоритме он движется только вперед и может уехать в обратном от источника света направлении.

Что подо мной

Последний датчик, который мы рассмотрим в данной статье – датчик линии. По сути, это тот же датчик света, но работаю-

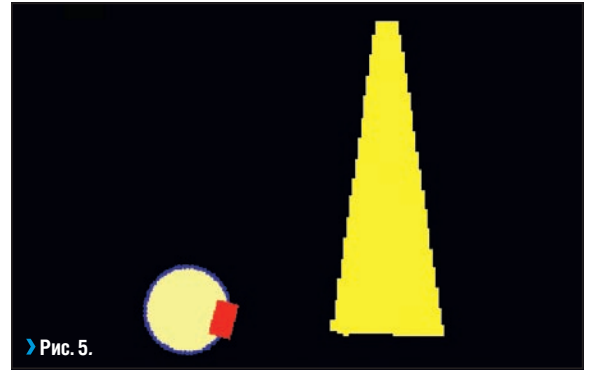


Рис. 5.

щий на отражение. Обычно датчик линии работает в невидимом инфракрасном диапазоне и позволяет определять, черная или белая поверхность находится под ним.

Алгоритм работы датчика линии аналогичен датчику касания, с той разницей, что сигналом будет не касание стенки, а нахождение черной линии, по которой должен двигаться робот.

Поэтому возьмите исходные данные от первого примера, замените поле, по которому будет двигаться робот, на поле с линией, и уменьшите размер датчика до небольшой точки (рис. 6).

Теперь модернизируйте алгоритм. В случае, когда робот не находит линию, он должен вращаться. То есть условие должно быть инверсным, и мы должны только повернуться на небольшой угол:

```

когда щелкнут по Robot
  повернуть в направление 0
  всегда
    идти 1 шагов
    если не цвет [касается ?]
      повернуться на 15 градусов
  
```

Увы, но наш робот может двигаться только по линии, которая имеет правые повороты. Для движения по линии произвольной формы нужно использовать два датчика, которые позволят также останавливать робота на горизонтальных черных линиях.

Видоизмените алгоритм самостоятельно, используя за основу рассмотренный ранее алгоритм движения робота с двумя датчиками касания, и заставьте робота ездить по произвольному пути.

На этом мы закончим наш мини-цикл. Вы можете узнать больше об робототехнике, использовании реальных микроэлектронных и робототехнических платформ Scratchduino, а также всю информацию, как самостоятельно построить их, методику обучения и многое другое на вики проекта Edumandriva по адресу <http://edumandriva.ru/wiki/index.php/ScratchDuino>. LXF

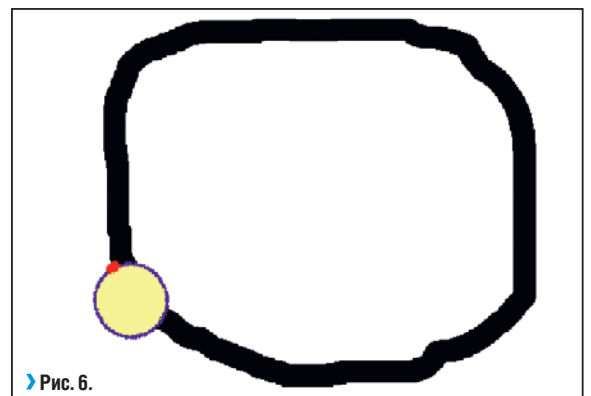


Рис. 6.

ПОДПИШИСЬ НА ЖУРНАЛ LINUX FORMAT!

2012

LINUX
FORMAT
Главное в мире Linux



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



Стоимость подписки Годовая — 2160 руб., полугодовая — 1170 руб. без учета стоимости доставки.

Адреса и телефоны редакции Санкт-Петербург, Лиговский пр., 50, корп. 15, тел. (812) 309-06-86.
Москва, Красноказарменная ул., 17, тел. (499) 271-49-54.

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

shop.linuxformat.ru

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 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ**Главный редактор**Кирилл Степанов info@linuxformat.ru**Литературный и выпускающий редактор**

Елена Толстякова

Переводчики

Илья Аввакумов, Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова

Редактор диска

Кирилл Степанов

Верстка, печатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Директор по рекламеВладимир Савельев advert@linuxformat.ru**Генеральный директор**

Павел Фролов

Учредители

Частные лица

Издатели

Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Ланпий»188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 4717**РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ****Редактор** Грэм Моррисон [Graham Morrison] graham_morrison@futurenet.com**Заместитель редактора** Эндрю Грегори [Andrew Gregory]agregory@futurenet.com**Штатный автор** Джонатан Робертс [Jonathan Roberts]jonathan.roberts@futurenet.com**Редактор диска** Бен Эверард [Ben Everard] ben_everard@futurenet.com**Художественный редактор** Эфраин Эрнандес-Мендосаefrain.hernandez-mendoza@futurenet.com**Выпускающий редактор** Гэри Уокер [Gary Walker] gary.walker@futurenet.com**Подготовка материалов** Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown],

Дэвид Хейворд [David Hayward], Грэм Моррисон [Graham Morrison], Джонатан Робертс

[Jonathan Roberts], Маянк Шарма [Mayank Sharma], Майк Сондерс

[Mike Saunders], Ник Вейч [Nick Veitch], Евгений Балдин, Андрей Боровский,

Артем Зорин, Татьяна Казанцева, Андрей Ушаков, Алексей Федорчук,

Александр Хрюкин

Художественные ассистенты Стейси Блэк [Stacey Black],

Кейт Мак-Доннелл [Kate McDonnell]

Иллюстрации Шейн Коллиндж [Shane Collinge], Крис Хедли [Cris Hedley],

Ely Walton Illustrations, iStockPhoto, Саймон Мидлвик [Simon Middleweek]

КОНТАКТНАЯ ИНФОРМАЦИЯ**UK:** Linux Format, 30 Monmouth Street, Bath BA1 2BWTel. +44 01225 442244 Email: linuxformat@futurenet.com**РОССИЯ:****Санкт-Петербург (редакция):**

Лиговский пр., 50, корп. 15

Тел. +7 (812) 309-06-86

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел./факс +7 (499) 271-49-54

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

LINUX – зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

«GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futurepic.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

В июньском номере

Надутый ЦРУ

Пусть вам и нечего скрывать – есть вещи, которые лучше держать при себе. Наше руководство поможет отвадить шпионов от вашего списка покупок.

Создать шрифт

Дизайн шрифтов – дело непростое; попробуйте сами и для смеха перекосите Comic Sans.

Сравнение: Альтернативные ОС

Если Linux для вас чересчур универсален, вам всегда остаются Haiku, BSD, BeOS, Syllable... и даже MikeOS!

Резвимся с GPS

Телефон на Android + приложение GPS + укрепления железного века = большая хохма, а потом можно еще и построить графики.

Содержание последующих выпусков может меняться без уведомления – мы сильно заняты: надо обрушивать снеговые горы.



Mandriva Linux — один из самых популярных дистрибутивов GNU/Linux в мире. Главные преимущества Mandriva — дружелюбный интерфейс, простота настройки, возможность быстрой адаптации пользователей, ранее не знакомых с этой ОС, совместимость с широким спектром программного и аппаратного обеспечения.

Корпоративные продукты Mandriva Linux

Mandriva 2010.2 Powerpack

Дистрибутив Mandriva 2010.2 Powerpack включает набор офисных и серверных приложений, и подходит для установки на офисной или домашней рабочей станции и на сервере. Дружелюбный интерфейс, простота настройки Mandriva Powerpack, совместимость с широким спектром аппаратного обеспечения и совместимость с «1С:Предприятие» обеспечивают корпоративным пользователям возможность легкого перехода с Windows на GNU/Linux.

Mandriva Enterprise Server 5

Mandriva Enterprise Server 5 (MES 5) — это надежный и производительный дистрибутив GNU/Linux для корпоративного сервера. MES 5 поможет вам снизить текущие расходы и упростить инфраструктуру. В MES 5 интегрированы серверные разработки программистов Mandriva, в том числе сервер каталогов пользователей Mandriva Directory Server, а также ведущие свободные серверные приложения, которые помогут вам с минимумом затрат времени и энергии настроить и поддерживать необходимые вам серверы. Срок поддержки дистрибутива — 5 лет.

Сертифицировано ФСТЭК

Дистрибутивы Mandriva Linux сертифицированы по требованиям ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ, что дает возможность использовать их для обработки конфиденциальной информации в автоматизированных системах класса до 1Г включительно и обработки персональных данных в информационных системах класса до К2 включительно.

- **Mandriva 2008 Spring Powerpack** — дистрибутив для рабочей станции или небольшого сервера.
- **Mandriva Corporate Server 4 Update 3** — дистрибутив для создания корпоративного сервера.
- **Mandriva Flash** — дистрибутив GNU/Linux, загружающийся и работающий прямо с USB-носителя.

EduMandriva — свободное ПО для образования

- Создано с участием российских преподавателей и методистов.
- Все ПО, необходимое для преподавания информатики.
- Методические материалы.

Наименование	Стоимость, руб.
Корпоративные продукты Mandriva	
Mandriva Linux 2010.2 Powerpack (DVD-Box)	1 300
Услуга подписки на Mandriva Enterprise Server 5 на 1 год, базовый уровень (с физическим носителем)	13 300
Услуга подписки на Mandriva Enterprise Server 5 на 3 года, базовый уровень (с физическим носителем)	34 800
Продукты Mandriva для образования	
Комплект программного обеспечения Mandriva Linux и EduMandriva для школ	3 500
Сертифицированные ФСТЭК продукты Mandriva	
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 10 рабочих мест	28 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 5 рабочих мест	14 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 1 рабочее место	4 990
Сертифицированный ФСТЭК Mandriva Corporate Server 4.0 Update 3	10 050

С вопросами по продуктам Mandriva обращайтесь в ГНУ/Линуксцентр!

MANDRIVA УЖЕ ИСПОЛЗУЮТ:
 МВД РФ, Минздравсоцразвития РФ,
 Минфин республики Саха (Якутия),
 Правительство Московской области,
 администрация Черниговского района,
 Приморского края, ОАО «Морион»,
 сеть магазинов «Компьютер-центр
 «КЕЙ», группа компаний «ИМАГ»,
 компания «Азбука мебели»,
 и многие другие.

Москва
+7 (499)

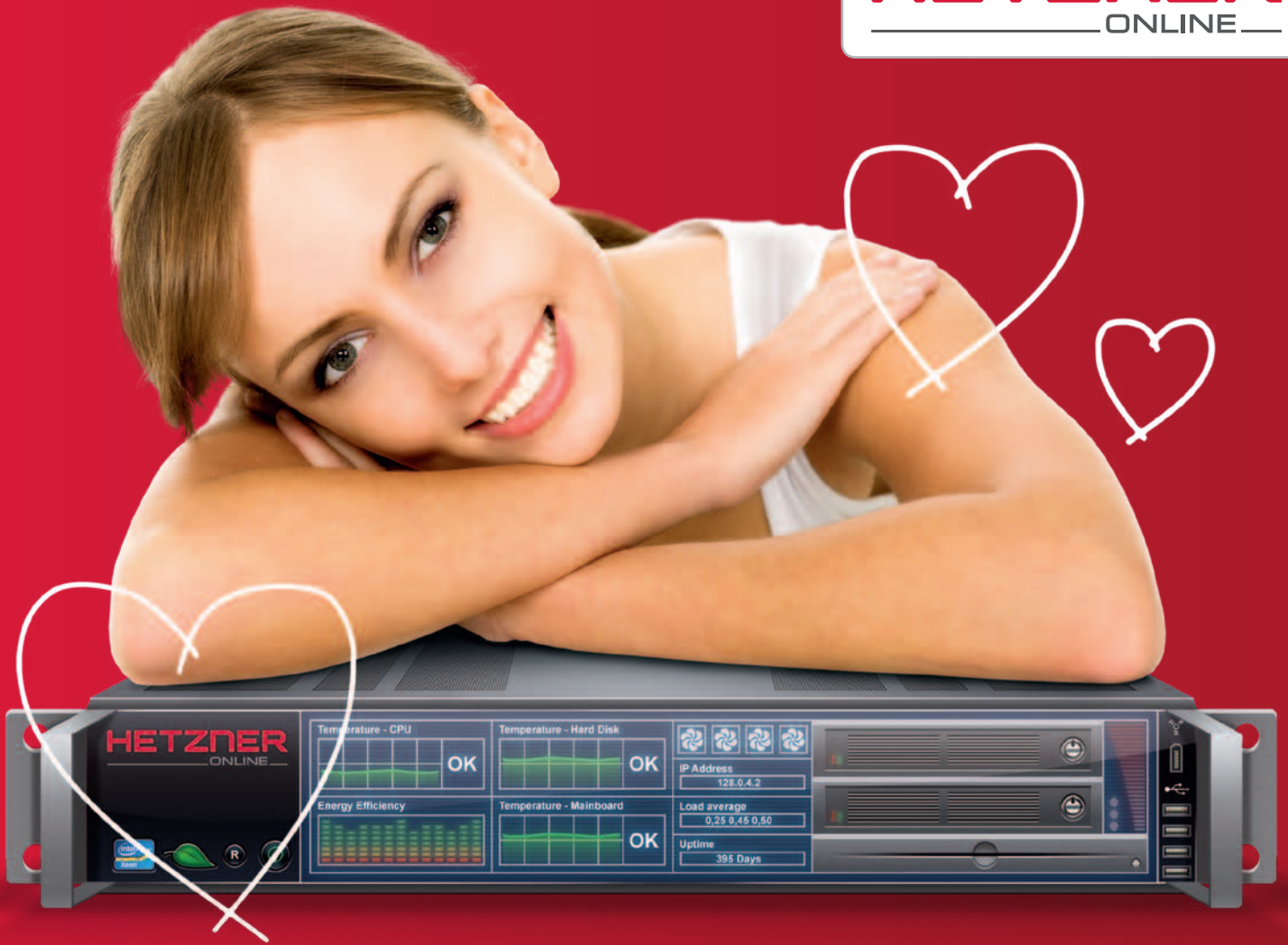
271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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





ВАМ ПОНРАВИТСЯ ЭТИ ПРЕДЛОЖЕНИЯ

HETZNER DEDICATED ROOT SERVER EX 4S

- Intel®Core™ i7-2600 Quadcore с поддержкой технологии Hyper-Threading
- 32 ГБ DDR3 RAM
- 2 x 3 ТБ SATA 6 Гбит/с 7200 об/м! (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 5500 рублей

2200
рублей в месяц

HETZNER DEDICATED ROOT SERVER EX 8

- Intel®Xeon® E3-1275 Quadcore с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM ECC
- До четырёх жёстких дисков на выбор
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- KVM-over-IP для постоянного доступа
- Установка из образов
- Без минимального контракта
- Установка 5500 рублей

от **3500**
рублей в месяц



GreenIT
Best Practice Award

2011

Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выберите более чистое будущее вместе с Hetzner Online!



RU.HETZNER.COM