

LINUX FORMAT



Внутри Fedora

Проникнем в мозг бравой бригады дистрибьютов с. 30

Главное в мире Linux

Август 2011 № 8 (147)

В ЭТОМ НОМЕРЕ

Советы по Linux

» Подробнейшие учебники

А вот и Mageia 1.0

» Мы подвергли тесту LXF общественную ветку Mandriva

Лучший подкаст для хакеров

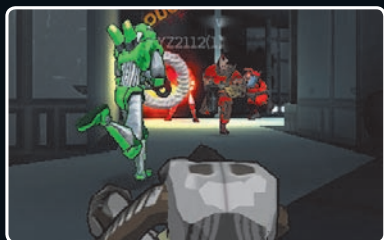
» ...слушать по дороге с работы

Фото по делу

» Свободное ПО может выдать профессиональное качество

Warsow: хрюшки с ракетометами

» Даже лучше, чем Crack Attack!



Вим Тейменс

« Писать код медиа-плеера было прикольно »

Да, простые были времена... с. 28



БИТВА ГИГАНТОВ!

Инновации или стабильность, простота или гибкость... какой дистрибутив — ваш?



Плюс!
Как это было

Продолжение саги о Linux с. 44



Ключевые навыки

Изучаем LXDE

» Дзен рабочего стола, или мир без треволнений

Хранение данных

Штурмуем облако

» Хранение данных локально — тренд уже устаревший

Настройка

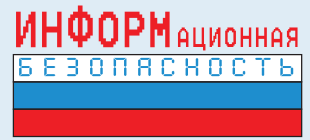
Дом для Gnome

» Дадим Gnome Shell элитную обстановку

ПОДПИСНЫЕ ИНДЕКСЫ В КАТАЛОГАХ
Агентство «Роспечать»: годовая подписка — 36343
полугодовая подписка — 20882
«Пресса России» — 87974, «Почта России» — 16572



10 ЮБИЛЕЙНАЯ ВСЕРОССИЙСКАЯ КОНФЕРЕНЦИЯ
«ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ.
Региональные аспекты. ИнфоБЕРЕГ-2011»



Важное деловое событие в области информационной безопасности
6-10 сентября 2011, Сочи

Организатор:
Академия Информационных Систем



10010010010001
ГОСУДАРСТВО
БИЗНЕС
БЕЗОПАСНОСТЬ
10212841752043987107593845

По вопросам участия обращайтесь:
Академия Информационных Систем,
г. Москва, ул. Первомайская, д. 126
Тел.: (495) 231-38-74, тел./факс: (495) 231-30-49
E-mail: conf@infosystems.ru
www.vipforum.ru, www.infosystems.ru

Регистрация участников до 1 сентября 2011

Информационная
поддержка:





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

Мы любим дистрибутивы, и в большой статье этого месяца рассмотрено шесть ярких примеров величия Linux. Мы спросили наших гуру: отчего ваш дистрибутив – особенный?



Майк Сондерс
Ответ простой: мой дистрибутив лучше, чем ваш. Т.ЧК. Конец дискуссии.



Эндрю Грегори
Мой TurboLinux строили официально лицензированные ниндзя. Ну нравится мне так думать...



Эфраин Эрнандес-Мендоса
В Ubuntu мексиканский диалект испанского – так и вижу себя с пивком «Корона» на солнышке.



Нейл Ботвик
Спец-ядро моего дистрибутива выводит сообщения на албанском. Типа: БУТ НИАСИЛИЛ АТТКЛЮЧКА ОС



Маянк Шарма
Вы знаете, мой дистрибутив – абсолютное чудо. Он уникален. Ему нет равных. Внимайте: в нем работает PulseAudio!



Джонатан Робертс
Мой дистрибутив стремится к миру во всем мире, включая KDE с Gnome и Vi с Emacs. Кхе... но все-таки не Nano.



Энди Ченел
А в моем есть setadp – обновления назад, и я магически путешествовую в прошлое, в эру Slackware 1.0.



Валентин Синыцын
Я сам собрал его из всяких запчастей. Нет, Неил, это не Gentoo.



Ник Вейч
Мой дистрибутив до того крут, что в хит-парадах имеет номер ноль. Да он вдобавок еще и невидимый.



Сюзан Линтон
Любя все дистрибутивы, я слила их в супер-дистрибутив под названием Openfedbuntiancore 2011,08 15 Ultimate Pro.



Шашанк Шарма
Мой дистрибутив лучше, чем у Майка. Реально, Майк, чего тут отрицать. Просто прими это как данность, ладно?



Боб Мосс
Шашанк и Майк, уймись или идите на улицу, на Linux-эквивалент парковки: comp.os.linux.advocacy.newsgroup



Польза сравнения

» В этом номере журнала мы предприняли очередную попытку сравнить дистрибутивы и выбрать лучший из них. Не могу не согласиться с британскими коллегами – любое подобное сравнение никогда не будет ни полным, ни объективным. Даже теоретически.

Представьте себе истинного коллекционера, пытающегося описать и классифицировать все существующие дистрибутивы. Уже к моменту завершения работы над первым томом их появится еще десяток. Масштаб проблемы наглядно проиллюстрирован на диаграмме <http://upload.wikimedia.org/wikipedia/commons/8/8c/Gldt.svg>. Так что о полноте исследования говорить не приходится.

С объективностью проблемы тоже возникают. Операционная система, как и любая программа, всего лишь инструмент для решения той или иной задачи. А задачи у всех разные. Инструмент же обычно подбирают не только «по задаче», но и «по руке» – в зависимости от субъективных ощущений удобства; а тут сколько людей, столько и мнений (это к вечному вопросу о «священных войнах» на некоторых популярных форумах).

Но сравнения все равно полезны – хотя бы только для того, чтобы поспорить с их автором... ну и еще раз подумать о собственном выборе. Задачи иногда меняются, представления об удобстве – тоже.

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

Закатываем рукава...

Обзоры

Mageia 1.0 10
Имя странное, но сообщество этого ответвления от Mandriva любит свое детище.

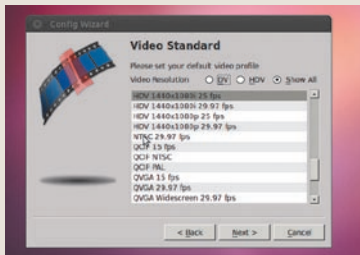


» Ощущается как старый друг...

FreeNAS 8.0 11
Новая версия свободной платформы для хранения данных – с поддержкой ZFS.

Warsow 0.62 12
Сражение с зомби в варианте для мягкосердечных умеренных пацифистов.

Kdenlive 0.8 13
Видеоредактирование – без труда. Ну, типа того – если одолеть интерфейс.



» Изобилие деловой начинки.

Сравнение: Подкасты

FLOSS Weekly 18
Going Linux 16
Hacker Public Radio 18
Linux Basement 16
Linux Outlaws 15
MintCast 17
The Linux Action Show 17
The Linux Link Tech Show 15

6 дистрибутивов
передрались
друг с другом
за ваши
симпатии
с. 20



БИТВА ГИГАНТОВ

ТРЕНИНГИ LINUX



Повысите квалификацию с Майком Сондерсом – повысят и вас с. 64



Что за штука – coreboot? с. 48

Люди говорят



« Windows 95 у меня жила дня два, до установки Linux. И я уже не оглядывался »

Вим Тейменс – про прошлое и будущее с. 28

На вашем бесплатном DVD

Fedora 15 и еще 8 дистрибутивов!

- » Полный Fedora 15 с новой Gnome Shell
- » ArchLinux, Mint, Debian, Ubuntu и openSUSE
- » Damn Small и Puppy Linux

ПЛЮС: Игры, инструменты, учебники... **с. 105**



Ищите в этом номере...

Внутри Fedora 30

Интересуетесь, как делают дистрибутивы? Мы пробрались за кулисы и все узнали.

Поправки к праву 34

Наши законодатели решили легализовать и без того вполне легальную GPL. Хотели как лучше?

20 способов угробить Linux 38

Linux будет попрочнее среднестатистической ОС – но и на него есть управа.

Приложения Android 40

Путеводитель по лучшим играм и умнейшим инструментам для вашего смартфона.

Пропустили номер?

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



Постоянные рубрики

- | | | |
|--|---|---|
| Новости 4 | Рубрика сисадмина 50 | Диск Linux Format 105 |
| Открыт сайт Национальной программной платформы, Интернет внедряется в быт Афганистана, Microsoft крохоборничает с патентами. | Невоспетый герой мира Linux: осваиваем Grub. | Содержимое двустороннего DVD этого месяца. |
| Сравнение 14 | Ответы 88 | Школа LXF 108 |
| Мы приложили ухо ко всем подкастам Linux, выясняя, который лучше. | ПРОБЛЕМЫ LINUX РЕШЕНЫ!
Оборудование отказало... дистрибутив долбанулся... Вам сюда! | Шустрые школьники давно с компьютером «на ты», но и учителя ушли от традиционного образа Марь Ивановны с указкой очень далеко... и даже удаленно. |
| Интервью LXF 28 | Hotpicks 94 | Через месяц 112 |
| Вим Тейманс и его GStreamer рвутся к мировому господству. | Отведайте горяченького: лучшие в мире новинки свободного ПО. | Протяните нам свою потную ладошку, и мы по ней погадаем: что-то будет? |
| Что за штука 48 | Пропустили номер? ... 104 | |
| Исследуем coreboot – возможный кандидат на замену BIOS. | Еще не поздно заказать любой из предыдущих выпусков журнала. | |

Учебники

Начинающим Дистрибутивы-легковесы 56

Ищете обезжиренный вариант? Боб Мосс нашел пару-тройку стройняшек.



» С Puppy Linux все просто.

Фотография Программы для профи 60

Серьезные требования к качеству фотографий – не повод отказываться от любимой ОС.

Образование Тренинги Linux 64

Школьные дни чудесные – спасибо Майку: с ним не страшен экзамен на сертификат. В этом месяце: лихо рулим процессами.

Сети Управление облаком 68

Организация файлов – это просто!

Командная строка Маленькие хитрости 72

Оказывается, командная строка способна забираться на сайты и добывать там новости.

Языки программирования Базовые сущности Erlang 76

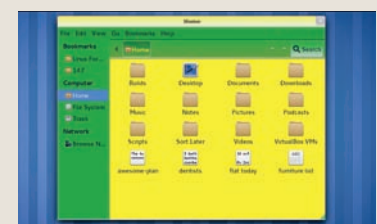
Изучаем фундаментальные строительные блоки для создания структур данных.

Языки программирования Библиотеки C++ 80

Библиотеками пренебрегать не рекомендуется: стоит ли плодить лишние велосипеды?

Hardcore Gnome Shell 84

Воюете с Gnome 3? Мы дадим вам средства настроить рабочий стол на ваш вкус.



» Хакерство начинается...



ГЛАВНОЕ Дело НПП сдвинулось с мертвой точки » Интернет добирается до глухих деревень » Microsoft зарабатывает на сутяжничестве

НОВОСТИ ИЗ РОССИИ

НПП: Действовать давно уже пора



» Рубрику готовил
ТИМУР МУБАРАКШИН

Национальная программная платформа наконец-то обзавелась собственным сайтом.

9 июня 2011 г. в московской гостинице «Золотое кольцо» прошла конференция «Свободное ПО: переход к реальным действиям»; Организаторами этого мероприятия выступили агентство маркетинговых коммуникаций CNews Conferences и CNews Analytics. На конференции, в частности, обсуждались следующие вопросы:

- » Каково место российских разработчиков на мировом рынке Open Source?
- » Насколько выгодно использовать открытое ПО в России?
- » Соответствует ли качество свободных продуктов потребностям рынка?
- » Какова динамика распространения Open Source в корпоративном и государственном секторах?
- » Насколько надежны и безопасны свободные программные продукты?

В рамках конференции состоялась дискуссия с участием экспертов в области свободного ПО, по итогам которой было сделано несколько интересных выводов.

Во-первых, по мнению экспертов, необходимо создать систему отбора, требований и стандартов, которая позволила бы обеспечить необходимое многообразие ПО. Под этим понимается привлечение максимально возможного количества разработчиков, решения которых дают возможность обеспечить гарантированную совместимость созданного ими СПО и целостность НПП. То есть, любая компания, которая соответствует требованиям, должна быть допущена к разработке НПП.

Во-вторых, важна интеграция отечественных разработок и с ведущими международными открытыми проектами, и с коммерческими продуктами крупнейших производителей. Немаловажным фактором успеха является взаимная заинтересованность всех участников этого процесса.

В-третьих, был высказан тезис о том, что необходимо обеспечить поддержку конечных пользователей СПО как на федеральном, так и на региональном уровнях. Конкретным шагом по его воплощению в жизнь эксперты считают создание Центров Компетенций (ЦК) по СПО во всех регионах страны. В рамках частно-госу-

«Лицензия на СПО значения не имеет, главное — результат.»

дарственного партнерства концерн «Сирius», куратор НПП, уже создал первые ЦК в Санкт-Петербурге, Кирове, Новосибирске и Самаре – и продолжает эту работу, пока за свой счет.

Также эксперты отметили необходимость внедрения открытого ПО и открытых стандартов в образовании, указав на проблемы такого внедрения, а также рассказав об «историях успеха», которые имеют место быть, несмотря на проблемы.

Отметим, что на конференции было указано, что лицензия на СПО в контексте НПП особого значения не имеет, главное – результат. Участники в этом уверены.

Новости короткой строкой

» Европейский Центр ядерных исследований (CERN) анонсировал выпуск обновленной открытой лицензии Open Hardware Licence 1.1, созданной с целью содействия обмену знаниями между различными сообществами разработчиков аппаратного обеспечения.
Источник: <http://press.web.cern.ch/press/PressReleases/Releases2011/PRO8.11E.html>

» Компания Skype, недавно купленная Microsoft, заявила, что отходит от былых принципов и готова предоставлять информацию о пользователях в ответ на запросы Федеральной службы безопасности. Кроме того, компания выразила готовность передать российским спецслужбам исходные тексты Skype и раскрыть используемые технологии и алгоритмы шифрования.
Источник: <http://www.gazeta.ru/business/2011/07/07/3688701.shtml>

» Представители правительства Бразилии подписали протокол о намерениях, в котором говорится о желании совместно работать с организациями Document Foundation и Apache Software Foundation над разработкой свободных офисных пакетов LibreOffice и OpenOffice.org.
Источник: <http://webmink.com/2011/07/06/brazil-signs-up-to-develop-office-suites/>

Альтернативную точку зрения на развитие СПО в России изложил генеральный директор «ПингВин Софтвер», вице-президент по продуктам, член Совета директоров компании Mandriva S. A. Дмитрий Комиссаров. По его мнению, СПО является в наши дни важным трендом, но «картина такого большого государства как Россия, всегда будет гетерогенной: речи о 100 % переходе на СПО быть не может». В связи с этим особый интерес представляет вопрос о порядке включения в репозиторий НПП проприетарного ПО.

Заместитель генерального директора ALT Linux Алексей Новодворский подчеркнул важность разработки профилей стандартов информационных систем в качестве меры для обеспечения совместимости различных типов ПО, а также необходимость создания публичных методик расчета полной стоимости лицензий на проприетарное ПО.

В ходе конференции состоялась дискуссия, на которой, среди прочего, обсуждались различные аспекты участия зарубежных производителей в проекте создания НПП, говорилось о необходимости наличия четко прописанных процедур для тестирования совместимости ПО, экспертно оценивалась реалистичность заявленных сроков и масштабов миграции на СПО, а также было проведено голосование. Подавляющее число участников предпочло концентрации усилий на развитии оригинальных отечественных инновационных технологий разработку СПО и решений, не имеющих соответствующих аналогов в России.

В конференции также принял участие Владислав Шершульский, руководитель программ технологического сотрудничества Microsoft в России. По его словам, со стороны Microsoft нет попыток ограничить распространение СПО, что звучит весьма странно, учитывая последние недружественные шаги Microsoft в отношении Android. Более того, особенностью прошедшей конференции стало спонсорство Microsoft.



► По словам Владислава Шершульского из Microsoft, данный софтверный гигант не препятствует распространению СПО.

В продолжение темы можно отметить появление сайта НПП, который «застолбил» себе сразу несколько похожих и труднопроизносимых доменных имен: tp-npp.ru, tp-nsp.ru, tpnpp.ru, tpnsp.ru, тпнпп.рф и тп-нпп.рф. Ресурс полностью посвящен технологической платформе и рассчитан на широкую аудиторию. Здесь будут освещены все этапы разви-

«На сайте будет площадка для обсуждения рабочих вопросов.»

тия НПП, а также освещены организованные в рамках проекта события и мероприятия. Участники НПП и желающие присоединиться к разработке платформы организации смогут получить информацию о порядке вступления, нормативные документы и прочие необходимые сведения. На базе сайта будет создана площадка для обсуждения рабочих вопросов, что позволит заинтересованным сторонам эффективнее взаимодействовать между собой.

Новости короткой строкой

► Евгений Додонов, руководитель по разработке Mandriva Linux, объявил об уходе из компании Mandriva. После реструктуризации компании в ноябре 2010 г. Евгений отвечал за разработку настольной Mandriva и курировал репозиторий Mandriva Cooker. Источник: <http://dodonov.net/blog/2011/07/>

► В рамках проекта nD создается портативная игровая консоль на базе Linux для детей от 6 до 12 лет. Планируемая цена – \$10–20. В продаже ожидается к концу текущего года. Источник: <http://www.linuxfordevices.com/c/a/News/nD-gaming-device/>

► Toyota Motor Corporation, крупнейшая японская автомобильная корпорация, объявила о вступлении в ряды некоммерческой организации Linux Foundation в качестве золотого партнера. Источник: <http://www.linuxfoundation.org/news-media/announcements/2011/07/toyota-joins-linux-foundation>

► Энди Рубин [Andy Rubin], вице-президент Google по разработке Android, сообщил о преодолении рубежа в 500 000 активаций устройств на основе Android в день, при росте распространения платформы на 4,4 % в неделю. Источник: <http://twitter.com/#!/Arubin/status/85660213478309888>

► Организация Creative Commons анонсировала выход книги «The Power of Open», с историями о людях, использующих лицензии Creative Commons в своей деятельности. Источник: <http://thepowerofopen.org/>

► Карен Сэндлер [Karen Sandler] назначена лидером организации GNOME Foundation взамен Сторми Питерс [Stormy Peters], перешедшей в компанию Mozilla Corp. Сэндлер работала генеральным советником в правозащитной организации Software Freedom Law Center, отстаивающей интересы СПО. Источник: <http://www.gnome.org/press/2011/06/karen-sandler-named-new-executive-director-of-the-gnome-foundation/>

► Из-за взлома временно закрыт сервис MtGox, в рамках которого работала одна из крупнейших бирж по обмену электронной валюты BitCoin на реальные дензнаки. Злоумышленникам удалось похитить базу пользователей MtGox – более 60 тыс учетных записей. Источник: <http://www.h-online.com/open/news/item/Bitcoin-exchange-closed-after-attack-1263448.html>

Благие намерения

Похоже, что дело НПП сдвинулось с мертвой точки. А именно, спустя несколько дней после конференции «Свободное ПО: переход к реальным действиям», ОАО «Концерн «Сириус», АРПП «Отечественный софт», НП «РАСПО» и НП «РУССОФТ» подписали соглашение о намерениях. Стороны обязались осуществлять взаимодействие по организации функционирования «Национальной программной платформы», создав для

этого одноименную некоммерческую организацию. Основной целью АНО «НПП» станет содействие развитию отечественной индустрии разработки программного обеспечения. Кроме того, стороны признали целесообразным включение в состав учредителей Российского союза ректоров и Российской академии наук, которые ранее подтвердили заинтересованность в официальной регистрации АНО «НПП».

ОТКРЫТОЕ ЖЕЛЕЗО

FabFi – Интернет в кишлаках

В Афганистане создается независимая от властей и цензуры сеть передачи данных. И местный горный рельеф этому только способствует.

Проект FabFi – это система с открытым исходным кодом, строящаяся при активном участии лаборатории FabLab и обеспечивающая беспроводную передачу сигнала Ethernet на расстояние по технологии Wi-Fi. По сути, это сеть беспроводных роутеров, созданная из самых обычных подручных материалов. По словам создателей, проект позволит в любой деревне построить свою скоростную беспроводную сеть передачи данных и предоставить жителям доступ к Интернету безо всяких ограничений и цензуры.

В данный момент сети на основе FabFi развернуты в двух странах, где доступ в Интернет затруднен в связи с тяжелой экономической и политической ситуацией и бедностью населения: Афганистане и Кении.

45 точек доступа FabFi развернуты в Афганистане в районе Джелалабада и еще в трех районах. Примечательно, что систему может расширить любой желающий: все необходимое для этого можно купить в ближайшем магазине. В результате конечная стоимость одной точки доступа составляет всего \$60.

Кенийские характеристики FabFi слегка отличаются от афганских в лучшую сторону, но в этой заметке мы коснемся именно Афганистана, потому что многие и представить себе не могут, что в стране, где война не прекращается уже 32 года, в стране, большая часть которой не контролируется никакими властями, и, наконец, в стране, где большинство населения неграмотно, может существовать свободный Интернет.

Проект FabFi стартовал в Афганистане в январе 2009 года, когда джелалабадское представительство FabLab продемон-



➤ Создатели самого протяженного участка сети FabFi Стив А. (справа) и Саид-Джала (слева) удовлетворены проделанной работой.

рировало возможности сети, бесплатно проведя высокоскоростной интернет в одну из соседних деревень, а также в университет, больницу и ряд неправительственных организаций Джелалабада. В основе FabFi – обычный маршрутизатор 802.11g, позволяющий передавать сигнал на расстояние не более нескольких десятков метров. Однако благодаря внешней антенне дальность передачи сигнала существенно увеличивается.

После месяца интенсивного обучения и практики работники джелалабадского представительства FabLab стали вполне компетентными в вопросе построения собственных сетей с нуля, и к ним выстроилась длинная очередь из представителей городов и деревень со всего Афганистана.

Наибольшее расстояние между точками доступа в Джелалабаде сейчас составляет 4 километра. И хотя максимальная пропускная способность сети составляет лишь половину от «номинала», обеспечиваемой стандартным маршрутизатором 802.11g (11,5 Мбит/сек против должных 22), система исправно работает при любых погодных условиях, будь то сильный дождь,

снег, смог или песчаная буря. Не помеха для сигнала и преграды вроде больших деревьев и стен. Отметим, что для строительства сети не используются какие-либо специальные мачты и башни: в качестве таковых выступает горный рельеф Афганистана и высокие постройки.

Вокруг FabFi в Афганистане уже выросло самое настоящее сообщество. Самая мощная точка доступа, в больнице Джелалабада, используется как основа для создания более мелких сетей, постепенно охватывающих все большую часть территории города. Открытость и дешевизна сети позволяет пользователям с легкостью общаться и обмениваться опытом, наращивая сеть. Сотрудничество является взаимовыгодным: группа пользователей FabFi в Афганистане получает ценные навыки строительства сети, а организация FabLab – дополнительный опыт построения беспроводных сетей в сложных условиях.

Сеть FabFi не стоит на месте: постоянно изменяясь, она совершенствуется как в технологическом, так и в социальном плане. В ближайшие месяцы создатели планируют повысить стабильность сети и дополнительно снизить ее стоимость, а также добавить новые возможности, увеличив при этом пользовательскую базу. Существуют планы по подключению локально расположенных в Афганистане ресурсов к глобальной сети и предоставления афганцам доступа к базе данных MIT OpenCourseWare – проекта Массачусетского технологического института по публикации в свободном доступе материалов всех курсов института. По мнению Fablab, это гораздо более ценно, чем простое увеличение пропускной способности FabFi.

Что такое FabLab

FabLab – это некоммерческая общедоступная мастерская, где можно изготовить почти все что угодно: от конфет до мобильных телефонов, от ювелирных изделий до средств транспорта, от пластмассовых игрушек до высокоточных инструментов. Это возможно благодаря набору станков с цифровым управлением – таких как принтер трехмерных объектов, станок для лазерной резки, фрезерный станок

и другие. Концепция разработана в Центре Битов и Атомов при Массачусетском Технологическом Институте (<http://cba.mit.edu/>). По сути, FabLab – катализатор творческого потенциала общества

FabLab предоставляет не только инструменты, но и модели для обмена знаниями, управления некоммерческими проектами и другие ключевые компоненты эффективного функционирования лабораторий.



➤ Так выглядят «антенны» передачи данных в сети FabFi. На снимке: две разнесенные на 4 км водонапорные башни с установленными на них передающими устройствами..

Разумные технологии для разумной планеты

Что означает «1,3 миллиона операций в секунду» для этого автомобиля

Это означает, что его потенциальный покупатель уже выявлен. Компания Asciom, глобальный поставщик маркетинговых услуг и технологий, совместно с IBM помогает девяти из десяти ведущих автопроизводителей, а также компаниям во всех крупных отраслях рационально использовать информацию из более 7 000 маркетинговых баз данных с тем, чтобы лучше ориентироваться в предпочтениях своих потребителей¹. В основе решения – серверы IBM System x[®] с процессорами Intel[®] Xeon[®], что позволяет компании Asciom консолидировать 9 360 разрозненных серверов в 264 системах eX5 без ущерба производительности². Разумный бизнес требует разумного программного обеспечения, систем и сервисов.

Сделаем планету разумнее. ibm.com/car/ru



Визуализация объема данных, отражающих склонность различных групп людей к покупке той или иной модели автомобиля.

¹ По данным, опубликованным на сайте IBM. ² По данным компании Asciom. Приведенный здесь пример иллюстрирует применение разработок IBM в указанной компании и потенциальные результаты этого. Экологические затраты и показатели производительности зависят от конкретного случая. Для получения информации о том, что IBM может сделать для вашей компании, свяжитесь с нами. Данные, смоделированные с целью визуализации эффективности товаров/услуг клиента, предоставлены компанией Asciom. Эти данные не включают в себя информацию о реальных потребителях или компаниях. IBM, логотип IBM, ibm.com, System x и изобразительное обозначение являются товарными знаками International Business Machines Corporation, зарегистрированными во многих странах мира. Наименования других компаний, продуктов и услуг могут быть товарными знаками или знаками обслуживания третьих лиц. Список товарных знаков, зарегистрированных IBM на настоящий момент, представлен по адресу www.ibm.com/legal/copytrade.shtml. Intel, Intel logo, Xeon и Xeon Inside являются товарными знаками либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран. © 2011 International Business Machines Corporation. Все права защищены.



**Мощный.
Интеллектуальный.**

ПАТЕНТНЫЕ ВОЙНЫ

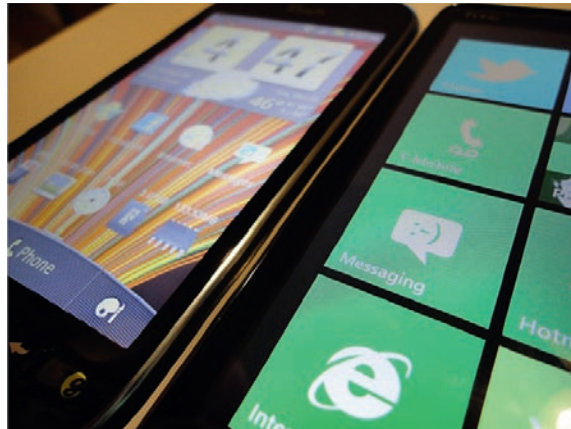
Microsoft идет в наступление

В последнее время компания начала последовательно «защищать» свои патенты. Особенно упорно гигант «защищается» от Android.

В течение месяца пришло несколько сообщений о патентных претензиях Microsoft. Сначала компания Стива Балмера объявила о подписании соглашения по сбору лицензионных отчислений за использование платформы Android в продукции компании General Dynamics Itronix. Список патентов, фигурирующих в сделке, не разглашается. Также не афишируется сумма, которую General Dynamics будет выплачивать Microsoft за каждое проданное устройство на базе платформы Android. General Dynamics Itronix – это небольшая канадская фирма, производящая защищенные ноутбуки и мобильные устройства для работы в экстремальных условиях.

Спустя некоторое время выяснилось, что Microsoft анонсировала заключение соглашения с компанией Velocity Micro, американским производителем компьютерной техники, согласно которому последний будет выплачивать Microsoft неафишируемую сумму лицензионных отчислений с продажи каждого планшета Cruz, построенного на базе Android. А буквально несколько часов спустя пришла еще одна новость – было объявлено о третьем за неделю соглашении по выплате отчислений за использование мобильной платформы от Google. На сей раз бумага была подписана с компанией Onkyo – японским производителем ноутбуков.

На этом наступление Microsoft не закончилось, и неделю спустя, на сайте компании появился анонс о заключении соглашения по выплате патентных сборов с тайваньской фирмой Wistron Corp. Соглашение затрагивает все производимые данной фирмой планшеты, мобильные телефоны, электронные книги и другие потребительские устройства, поставляе-



» В ближайшее время основная борьба за лидерство на рынке мобильных ОС развернется именно между Android и Windows Phone 7. Благодаря нечестной политике Microsoft, WP 7 может и выиграть.

мые с прошивками на основе платформ Android и ChromeOS. Сумма, которую будет выплачивать Wistron Corp, список фигурирующих патентов и другие детали сделки не разглашаются.

Общественность увидела в этом наступлении на мелких и средних производителей компьютерной техники под-

«Сделка приносит Microsoft больший доход, чем продажа ОС.»

готовку плацдарма для атаки на более крупные фирмы. Во всяком случае, связанная с Android сделка с HTC уже приносит Microsoft больший доход, чем продажа собственной ОС Windows Phone. С учетом отчислений \$5 за каждый проданный телефон получается, что HTC уже передала Microsoft 150 млн долларов, в то время как Microsoft удалось продать только 2 мил-

лиона лицензий на Windows Phone, по цене \$15 за лицензию, и общий объем вырученных средств равен 30 млн долларов.

Опасения общественности оправдались довольно скоро – уже через несколько часов появилось известие о том, что Microsoft потребовала у Samsung выплаты лицензионных отчислений за использование своей интеллектуальной собственности в платформе Android, в размере 15 долларов за каждый произведенный смартфон. Для справки: только во втором квартале 2011 года Samsung выпустила около 19 млн смартфонов, завоевав вторую позицию в рейтинге производителей. Первое место удерживает компания Nokia, но Samsung имеет все предпосылки, чтобы занять место лидера. Дебютировавший в конце мая смартфон Galaxy S II, работающий на базе ОС Android, набирает популярность: за следующий месяц было продано около 3 млн экземпляров.

Из компаний, которые не согласились выплачивать Microsoft отчисления и были таким образом вовлечены в судебное разбирательство, можно отметить Motorola и Barnes & Noble.

Похоже, Microsoft пытается «не мытьем, так катаньем» добиться своего. Потеряв лидерство на рынке мобильных операционных систем, редмондский гигант, прикрываясь мифическими патентами, пытается подрывать благосостояние открытого ПО – в частности, платформы Android.

Между тем, один из старейших американских телекомов, Nortel, заявил, что его патенты будут проданы консорциуму, который объединил Apple, EMC, Ericsson, Microsoft, Research In Motion и Sony. Консорциум предложил за портфель патентов Nortel значительно более высокую цену, чем Google – 4,5 миллиарда долларов. Находясь в стадии банкротства, компания Nortel выставила на продажу более 6 тысяч патентов, охватывающих почти все области, связанные с телекоммуникациями. Портфель патентов Nortel стоимостью 900 млн долларов надеялась выкупить компания Google, которая хотела сформировать на базе этого портфеля своего рода щит для отражения патентных атак не только против себя, но и против партнеров и сообщества разработчиков открытого ПО, вовлеченных в работу над Android и Chrome.

Вторая попытка

В прошлом Microsoft уже предпринимала попытки собирать дань с производителей СПО. В 2007 году компания Microsoft заявляла, что в Linux и открытом ПО нарушено 235 патентов. Указанные патенты так и не были поименованы, а все связанные с ними сделки проводились с обязательной подпиской о неразглашении. Благо-

даря этим патентам, Microsoft удалось заключить соглашения с компаниями Novell, Fuji-Xerox, Samsung и Xandros. Компании Mandriva, Canonical и Red Hat отказались подписывать соглашение с Microsoft, и никаких судебных разбирательств после этого не последовало. Похоже, что сейчас ситуация куда более серьезная.

В этом выпуске новостей использованы материалы с сетевых ресурсов www.sputnik.ru, www.snews.ru, www.fablab.af, www.microsoft.com и др.



АЛЕКСЕЙ ФЕДОРЧУК
...простой советский юзер.

Linux в «верхнем» образовании

О СПО в школьном образовании нынче не пишет только такой ленивый, как автор этих строк. Но применительно к образованию высшему эта тема затрагивается гораздо реже. Хотя именно здесь СПО самое место. И не только для IT-специальностей, но и для любых инженерных и естественнонаучных.

Пробел восполнила интересная конференция, организованная порталом nixp.ru в конце мая. Затрагивались вопросы от коммерциализации СПО до web-разработок на его базе, но в связи с темой данной колонки я остановлюсь на докладе Дмитрия Шурупова «Свободное ПО в образовательном процессе кафедры ИКТ МИЭМ».

Сам доклад ищите на nixp.ru; вкратце — Дмитрий описал фактически самодостаточную образовательную систему, исполняющую также роль социальной сети. Причем созданную самими студентами в ходе обучения.

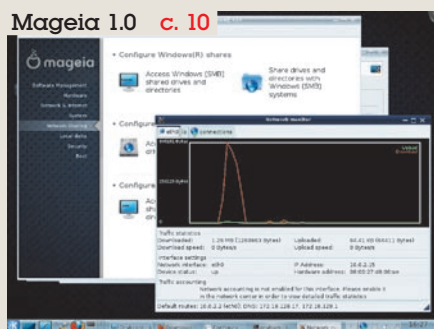
Увы, это грозит стать и слабым звеном системы: студенты защищают диплом и уходят. А притока новых кадров может и не быть. И это общая болезнь СПО — не только в образовании и не только у нас: перворазработчики уходят в иные сферы, а на смену им либо не приходит никто, либо приходят — и перекаивают все по своему. Чтобы затем тоже уйти.

Разработка СПО — не хобби. Это такая же профессия, как у научного работника: без немедленной практической отдачи. И возможность профессиональной работы в сфере СПО требует сочетания массы факторов. До сих пор, кроме единичных случаев, оно было реализовано только в героические времена разработки BSD.

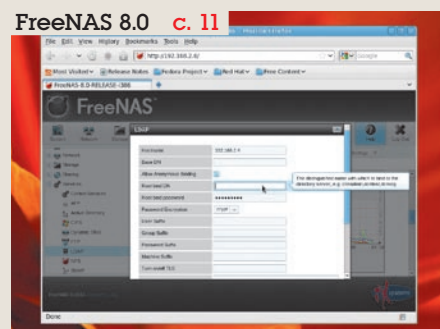
alv@posix.ru

Сегодня мы рассматриваем:

- Mageia 1.0** 10
Данное ответвление от Mandriva, созданное сообществом, заставило нас гадать: как произносится его название, и хорошо ли это? А теперь посмотрим, реальное оно конкурент или так, сбоку припека.
- Warsow 6.7** 12
Если вы вечно проигрываете в *Quake*, поскольку в потемках не разобрать, что происходит, но любите палить из гранатомета, не отчаивайтесь: в *Warsow* вы отведете душу, давая залпы по вражеским свиньям.
- FreeNAS 8.0** 11
Чуть ли не каждую пару месяцев нам присылают устройство NAS для раскурочивания; на этот раз, благодаря FreeNAS, мы заполучили еще и специализированную серверную версию BSD для установки на нем.
- Kdenlive** 13
Средства для профессионального редактирования видео в Linux имеются; не обижены и домашние пользователи. *Kdenlive* претендует на звание золотой середины — причем небезуспешно.



➤ Рабочий стол построен на KDE 4, но выглядит несколько более традиционно, чем в других дистрибутивах.



➤ Похоже, все устройства NAS зашли на Debian, но экземпляр по версии BSD тоже стоит рассмотреть.

Наш вердикт: Пояснение

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



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

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

LINUX FORMAT Вердикт

Spotify

Разработчик: Spotify Ltd
Сайт: www.spotify.com
Цена: Бесплатно/\$4,99/£9,99

Функциональность	9/10
Производительность	10/10
Простота использования	9/10
Оправданность цены	10/10

➤ У любителей музыки Spotify действительно может вызвать слезы счастья — настолько он хорош.

Рейтинг 10/10

Mageia 1.0

Грэм Моррисон признает, что «общественная» ветка Mandriva не лишена магии... хотя затрунდება в произношении названия дистрибутива.

Вкратце

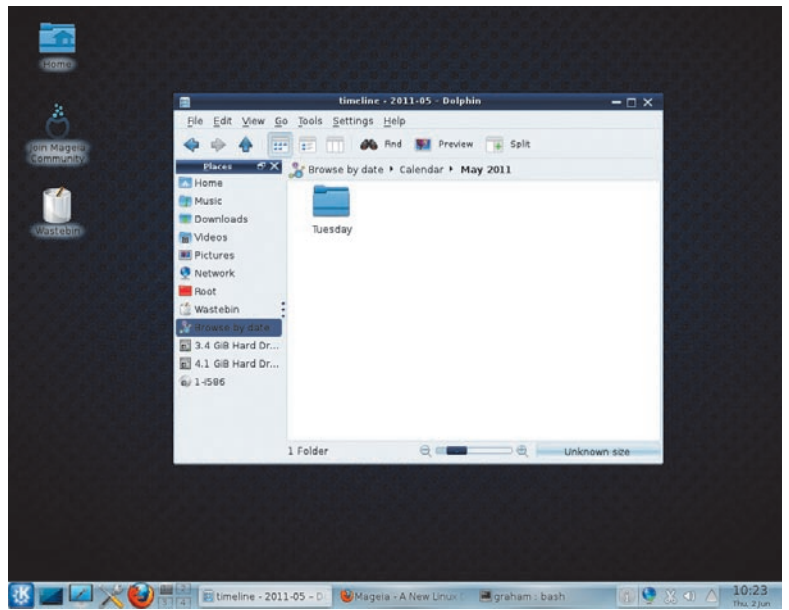
» Дистрибутив на базе KDE, ответивший от Mandriva и поддерживаемый сообществом пользователей. См. также: сам Mandriva или Kubuntu.

Сначала – немного истории. Mandrake был в Linux одним из первых дистрибутивов, не требующих ученой степени для их установки. Влившись в 2005 году в фирму Connectiva, дистрибутив сменил имя на Mandriva. Финансовые трудности 2010 года привели к созданию Mageia – его основали несколько бывших сотрудников компании и членов сообщества. Mageia 1.0 вышел лишь за несколько недель до основной версии Mandriva. Однако новых функций в нем не предусмотрено (во всяком случае, пока): версия 1.0 лишь оформила переход из корпоративного в общественный статус и создала задел на будущее. Установив Mageia, вы обнаружите только аккуратный ребрендинг.

Простота установки всегда считалась сильнейшей стороной Mandriva – и Mageia унаследовал эту особенность, хотя и под новой «вывеской». Выглядит все фантастично, и через какие-нибудь 30 минут, ответив на несколько вопросов,

«Уровень дистрибутива, созданного сообществом, вызывает восторг.»

вы уже в рабочем столе. Приятно, что выбор рабочего стола остается за пользователем, а не навязывается, и мы предпочли KDE. Mageia 1.0 укомплектован новейшим KDE, 4.6.3, но в заботе о спокойствии пользователей Gnome ограничились стабильной версией Gnome 2.32. Для первой



» Mageia работает на KDE 4 – и не вызывает желания вернуться на Gnome или Windows.

версии – шаг оправданный, но в планах – переход на Gnome 3.0, и нам не терпится увидеть, как разработчики расправятся со стандартным интерфейсом Gnome.

Раздвоение десктопа

KDE 4 от Mageia воздержался от ряда скандальных функций. Виджетов по умолчанию нет, а меню запуска напоминает KDE 3.x. Фон рабочего стола преднастроен как файловый менеджер, а не коллекция виджетов, и выглядит гладким, знакомым и отзывчивым.

Как и в большинстве дистрибутивов с KDE, демон фоновой поиска по умолчанию не активирован, что вызывает замешательство при поиске по дате в файловом менеджере *Dolphin*; активируйте демон, перейдя в окно системных настроек.

Знакомая панель управления Mandriva теперь называется Mageia Control Center [Центр управления Mageia]. Добавить пользователей, настроить общий доступ по NFS и брандмауэр и установить *Compiz* было делом нескольких щелчков. Однако здесь необходимо кое-что поправить. Например, общей панели не помешает поле поиска, а интерфейс пользователя порой теряет связь с фоновым рабочим столом.

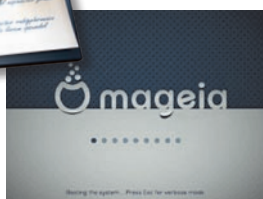
Другая часть ценного наследства – управление пакетами, и команда Mageia

проделала огромную работу по заполнению своих репозиториях тысячами приложений. Новейшие версии многих программ входят в стандартный комплект, в том числе *LibreOffice* и *Firefox 4*.

За исключением обновления пакетов и ядра, между Mageia и старым Mandriva различий не так уж много, однако столь высокий уровень дистрибутива, созданного сообществом, вызывает восторг, а команда достойна наивысших похвал за доблестный труд. С нетерпением ждем дальнейших успехов Mageia. **LXF**



Свойства навскидку



Отличная графика

Художники поработали блестяще, буквально преобразив старый интерфейс.



Удобство пользования

Mageia взял от Mandriva надежную поддержку и простоту в использовании.

LINUX FORMAT Вердикт

Mageia 1.0

Разработчик: Сообщество Mageia
Сайт: www.mageia.org
Лицензия: GPL

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

» Достойный – и многообещающий – наследник Mandriva.

Рейтинг **8/10**

FreeNAS 8.0

Очередная инкарнация – водораздел для сетевых хранилищ с открытым кодом? Шашанк Шарма видит явный уклон к предприятиям.

Вкратце

» *FreeNAS* – бесплатный сервер для сетевого хранилища данных на базе FreeBSD. Предоставляет web-интерфейс пользователя, поддерживает ZFS, доступ к файлам с компьютеров под Windows, Mac и Linux и многие другие функции. См. также: *Openfiler*.

Единственная задача файлового сервера – обеспечить доступ сетевых компьютеров к файлам и документам. Сервер NAS, контролирующий сетевое хранилище данных, реализует эту задачу, а также упрощает административную работу и ускоряет доступ к данным. *FreeNAS*, сервер NAS на основе FreeBSD, служит для обмена данными между машинами Mac, Windows и Linux.

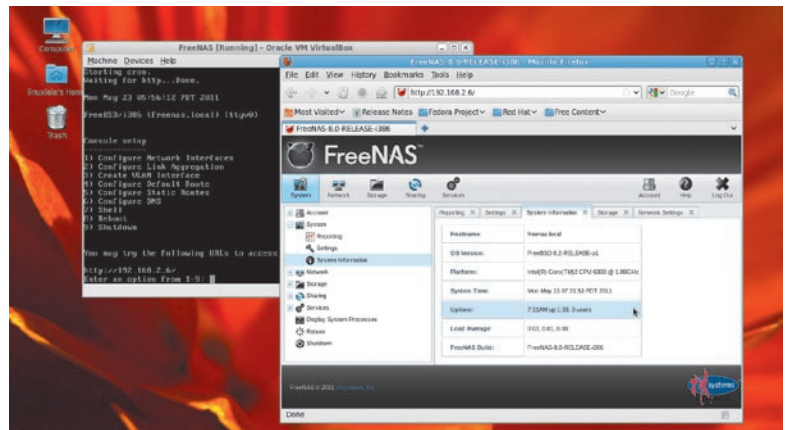
Версия 8.0 обозначила смену курса *FreeNAS*. Самое заметное новшество – переход на «работающий образ». Прежние версии позволяли хранить данные на том же диске, где был установлен сервер, а *FreeNAS 8.0* требует себе специальное устройство.

Причина такой перемены в том, что современные устройства *FreeNAS* состоят из множества секций, каждая из которых может содержать несколько работающих образов. Если сервер находится на отдельном устройстве, обновить образ или откатить его на старый гораздо проще.

«Комплект функций приближает ZFS к идеалу — но и требует ОЗУ...»

Поэтому при установке *FreeNAS* на жесткий диск сервер займет его полностью, независимо от объема. Идеальное устройство для установки *FreeNAS* – флэш-накопитель.

Собственно установка не требует много дисковой памяти и загружается за считанные секунды. На этом этапе сервер дол-



» Исследуйте интерфейс — просмотрите меню на боковой панели и кнопки вверх; но первым делом смените пароль!

жен выдать сетевой адрес; если этого не произошло, придется настроить сетевые карты вручную. Наберите полученный адрес в браузере, и перед вами откроется графический интерфейс настройки. Войдите с именем пользователя `admin` и паролем `freenas`.

Сервер *FreeNAS* работает с файловой системой UFS, но версия 8.0 поддерживает и ZFS 14. Расширенный набор функций, таких как общий доступ к томам по протоколу CIFS, приближает эту ФС к идеалу – но и требует ОЗУ объемом не менее 6 ГБ.

При испытании *FreeNAS* с ZFS на Fedora 14 (4 ГБ ОЗУ) система работала чрезвычайно медленно, а спустя какой-то час окончательно зависла. Требования UFS к ОЗУ значительно скромнее, и для рядовой машины лучше выбрать эту ФС.

Должные инструменты

Как и любой сервер для NAS, *FreeNAS* работает с NFS, SMB, CIFS и другими протоколами передачи файлов. Поэтому на сетевые машины придется установить инструменты, соответствующие операционным системам, например *nfs-utils* и *autofs*.

FreeNAS поддерживает *rsync*, что упрощает резервное копирование со снимком состояния системы. Периодически выполнять такие снимки чрезвычайно полезно, если вы боитесь утраты данных. Однако снимки, сжатие и квоты доступны на ZFS, но не на UFS – помните об этом.

Полный перечень функций здесь не поместится – просто скажем, что в *FreeNAS* есть все, что нужно для сетевого накопи-

теля, и даже больше. Однако подобное богатство оборачивается усложнением интерфейса. По этой причине разработчики применили вертикальное разворачивающееся меню – оно быстрее доставит вас на место, чем навигационные вкладки.

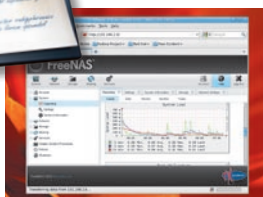
На время написания статьи документацию с версии 8.0 (вышедшей месяц назад) еще не обновили, но каналы IRC, списки рассылки и форумы – хороший источник информации, и ответ на любой возникший вопрос найти легко.

Поскольку сервер основан на FreeBSD, поддержка аппаратного обеспечения (сетевых карт и пр.) зависит от FreeBSD 8.0.

Если отвлечься от досадных мелочей, то можно смело утверждать, что вы видите перед собой программный продукт высшего класса. **LXF**

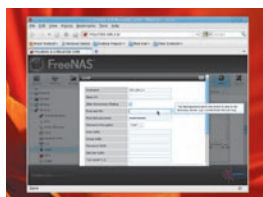


Свойства навскидку



Создание отчетов

Выбор команды System > Reporting [Система > Отчеты] дает доступ к массе полезных графиков.



Поддержка LDAP

Клиент LDAP упрощает авторизацию пользователей для доступа к данным.

LINUX FORMAT Вердикт

FreeNAS 8.0

Разработчик: Спонсируется компанией iXsystems
Сайт: www.freenas.org
Цена: Бесплатно по лицензии BSD

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

» NAS вы, считайте, нашли. Мощное сообщество, толковые разработчики... качество будет только повышаться.

Рейтинг 8/10

Warsow 0.62

Необычная, но все же – стрелялка... сумеет ли ее обличенность убедить миролюбца **Джонатана Робертса** втянуться в кровавый жанр?

Вкратце

» Стрелялка от первого лица с уникальной интерпретацией классической формулы. См. также: *Red Eclipse*, *Nexuiz* или *Tremulous*.

Кровь, море крови. И много стволов. Необычные, чуждые пейзажи. Долгие годы все это было на знамени жанра стрелялок от первого лица [FPS, First Person Shooter] – они никогда мне не нравились. Зачем же я взялся за обзор *Warsow*, игры с явными признаками *Quake* и *Unreal*?

Во-первых, несмотря на их влияние, *Warsow* все же не обычная FPS. Игра основана на QFusion, модернизированном игровом движке *Quake II*, но избегает угнетающе мрачной графики, характерной для насыщенных насилием, жестоких игр данного жанра.

Вместо этого разработчики обратились к мультяшному стилю (этот метод, так называемый сел-шейдинг, часто встречается в комиксах Manga, а впервые применен для компьютерных игр в *Jet Set Radio*).

Немного легче

Влияние этого решения поразительно, и тут же бросается в глаза. В целом игра «легче» средней FPS, а резкий контраст отчетливее выделяет предметы и соперников и упрощает прохождение карт, поэтому новичкам будет в ней просто.

Вдобавок мультяшный стиль умеряет насилие. Бесспорно, ни одна FPS не обходится без пистолетов и гранатометов – но крови здесь нет, а поражение противника оставляет ощущение скорее победы в соревнованиях стрелков, чем садистского убийства.



» Мультяшный стиль *Warsow* дополнен репликами из комиксов.

Игровой процесс *Warsow* подобран в том же русле. Ключевые элементы *Quake* и *Unreal* (прыжки, подскоки и броски различных видов) используются широко, и овладение ими – залог успеха.

Например, пройти карту на дальнейшее расстояние быстрее всего не бегом, а комбинацией бросков и прыжков. Во многих режимах скорость решает все, поскольку позволяет собрать все бонусы раньше соперников.

Это побуждает игроков изучить более широкий диапазон навыков, прежде чем они ощутят себя асами. К счастью, система движений хорошо документирована, а разработчики постарались упростить доступ к ним, «навесив» основные элементы на одну горячую клавишу.

Системе движений даже отведен отдельный игровой режим: соревнования игроков на скорость прохождения трассы. В этом режиме можно упражняться вне сети или воспользоваться *Racesow*, сервером на сайте www.warsow-race.net. Здесь вы всегда найдете компанию желающих померяться силами (более 2000 человек ежедневно).

Racesow – только одна из ячеек чрезвычайно активного сообщества, окружающего *Warsow* в Сети. На главных серверах постоянно идет сразу несколько игр,

среди которых всегда найдется подходящая по набору режимов, карт и количеству игроков. Кроме того, можно вступить в одиночку и оставить запись о победе (ну, или поражении) потомкам.

Сходство *Warsow* с традиционными FPS делает игру превосходным тренировочным полигоном для новичков в этом жанре. Притом, несмотря на специфическое решение, стрелялка достаточно притягательна и для стрелков-ветеранов. Словом, попробовать *Warsow* несомненно стоит. **LXF**

Прыжки по стенам!



Движение – центральный элемент *Warsow*. Хотите победить – осваивайте прыжки по стенам.

LINUX Вердикт
FORMAT

Warsow 0.62

Разработчик: Warsow Development Team
 Сайт: www.warsow.net/
 Цена: Бесплатно на условиях GPL

Сюжет	8/10
Графика	8/10
Увлекательность	9/10
Оправданность цены	10/10

» Превосходная FPS упрощает новичкам и слабонервным знакомство с жанром.

Рейтинг 8/10

Kdenlive 0.8

Энди Ченел придвинул свое директорское кресло и вступил в революцию YouTube, прихватив новую версию видеоредакторского комплекса для Linux.

Вкратце

» Kdenlive — видеоредактор для Linux, OS X и FreeBSD. Можно попробовать также OpenShot или Avidemux.

Накопители раздуваются, носители плодятся... еще несколько лет назад кто бы подумал, что на цифровой камере поместится четырехчасовая речь товарища Фиделя и футбольный матч впридачу! При таком обилии отснятого материала без приличного редакторского ПО не обойтись.

Kdenlive 0.8 доступен для Linux, OS X и FreeBSD, но возможность установки двойного пакета зависит от дистрибутива.

На небольшом экране интерфейс выглядит слегка перегруженным, но при высоком разрешении (при котором и обрабатывают видео) все отлично. Можно создавать и сохранять собственные раскладки, и при переходе с *Premiere* или *Final Cut* нетрудно скомпоновать похожий интерфейс.

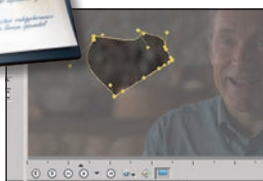
Предусмотрена возможность захвата видеоматериалов с ленточных накопителей и карт SD; разрешение можно выбрать любое, от стандартного определения PAL до 1080p.

«Сложность его хотя и не непостижима, но выглядит таковой.»

Базовые операции работают прекрасно, а перемещение клипа по монтажной линейке и — в особенности — добавление переходных эффектов выполнены элегантно. Что способствует ускорению и упрощению видеомонтажа, причем тонкая регулировка с укрупненным масштабом монтажной линейки тоже сохраняется.

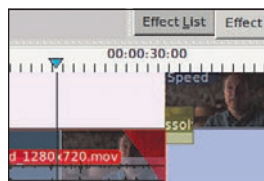


Свойства навскидку



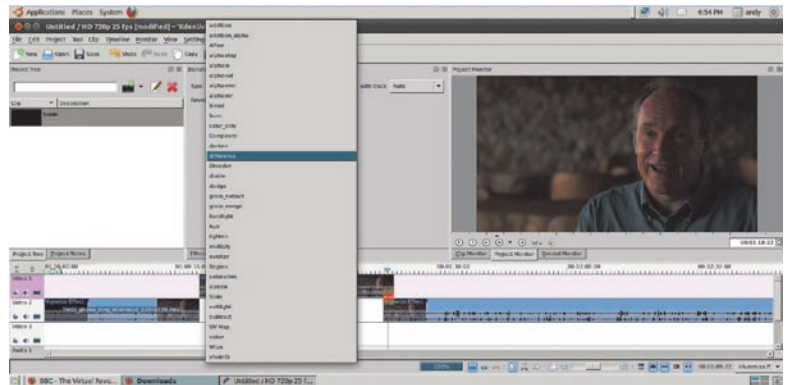
Ротоскопирование

Позволяет создавать сложные видеомаски с помощью кривых Безье.



Интерфейс

Интуитивно понятный интерфейс значительно упрощает добавление переходов и видеоэффектов.



» С ростом разрешения и усложнением материала Kdenlive начинает буксовать.

Разработчики Kdenlive добавили несколько функций, в том числе такой тяжеловес, как ротоскопирование (создание анимации по опорным кадрам с помощью кривых Безье) и менее требовательную светографику.

Правда, не все эффекты идеальны. Например, при изменении скорости воспроизведения звуковая дорожка просто пропадает — а не ускоряется (с неизбежной сменой тона) или растягивается, сохраняя синхронность и тон. Но если вы разделите звук и видео, синхронность и тон уцелеют (хотя и с ущербом для быстродействия). Светографика — забавная, но сложная функция, и лучше оставить ее для рекламных фото.

Пауза в воспроизведении

Интервал между нажатием клавиши пробела и остановкой воспроизведения заставляет ругнуться на экран. Это было проблемой более ранних версий; теперь ситуация улучшена, но по-прежнему, чем больше вы втягиваетесь в работу, тем хуже быстродействие.

Kdenlive — не простой редактор по типу *iMovie*. Но увеличение мощи неизбежно влечет усложнение; а возможность наложить на видеоконтент стопку неструктивных эффектов хотя и способствует свободе творчества, но довольно сильно напрягает оборудование.

По мере роста разрешения, частоты кадров и сложности редактирования ПО начинает «тормозить». Правда, на нашем тестовом ноутбуке с процессором Core i5 и 4 Гб ОЗУ в формате 720p при раз-

решении 1440×1080 работа шла вполне приемлемо (правда, на мониторе предпросмотра отмечался сброс кадров для сохранения синхронности звука и видео). Естественно, чем больше эффектов добавить к исходному материалу, тем заметнее сброс кадров.

Нам понравились методы комбинирования эффектов и сохранения удачных настроек. Опция сохранения всей конфигурации послужила бы логическим завершением этого ряда.

Основная проблема Kdenlive, вероятно, в том, что он не дотягивает до уровня профессионалов, а для случайного пользователя сложноват. Сложность эта хотя и не непостижима, но выглядит таковой, если вам некогда вдаваться в изучение. Однако при потребности в сложных задачах и азарте к обработке Kdenlive — неплохой выбор. **LXF**

LINUX FORMAT Вердикт

Kdenlive 0.8

Разработчик: Kdenlive
Сайт: www.kdenlive.org
Цена: Бесплатно на условиях GPL

Функциональность	8/10
Быстродействие	6/10
Удобство использования	7/10
Документация	5/10

» Программа достойная, но входной интерфейс может отпугнуть неопытного пользователя.

Рейтинг 6/10

Сравнение

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

Подкасты Linux

Лавируя в напряженном трафике, **Маянк Шарма** бороздит просторы Internet в поисках самых увлекательных и информативных подкастов Linux.



Про наш тест...

Чтобы сузить диапазон поиска, мы отобрали только подкасты, посвященные Linux и Open Source, проигнорировав ряд блестящих шоу про технологии, лишь изредка касающихся нашей темы. Очевидно, главное для нас — контент; шоу с интервью сложны в производстве и не особо привлекают, разве что в гостях супер-знаменитость.

Благодаря доступности высококлассной звукозаписи, аудиокачество — не проблема, если только хосты не разнесены территориально, как часто бывает с подкастами.

Мы считаем, что хороший подкаст должен втягивать аудиторию, и шоу, соответствующие этому критерию, получают лишние баллы.

Наконец, долговечность проекта сама по себе достоинством не является. Но зачастую шоу, демонстрируемые годами, имеют больше шансов продолжаться, поскольку обросли лояльной аудиторией слушателей.

Наша подборка

FLOSS Weekly c. 18
Going Linux c. 16
Hacker Public Radio c. 18
Linux Basement c. 16
Linux Outlaws c. 15
MintCast c. 17
The Linux Action Show c. 17
The Linux Link Tech Show c. 15

В идеальном мире вы не слушали бы ничего, кроме подкаста LXF на www.tuxradar.com, но, благодаря закону многообразия, можно настроиться на кучу трансляций и регулярно получать порцию новостей из мира Linux, различных точек зрения и интервью.

С момента нашего сравнения подкастов в 2007 году мир Linux изменился весьма радикально. Тогдашний победитель, LUG Radio, как и пара других отмеченных в обзоре подкастов, выходить перестали, но вместо каждой из отошедших в прошлое трансляций появилась едва ли не дюжина новых.

Во многом этот взрыв обязан техническому прогрессу: если в 2007-м создать подкаст было просто, то сейчас это элементарно.

Изменилось и содержимое подкастов. Раньше они в основном ограничивались новостями, а сегодня большинство ведущих делятся своими мнениями. Пусть вы с ними не согласны, но нередко именно личные точки зрения и отличают один под-

каст от другого. Одно подкасты транслируют интервью с людьми-моторами и сотрясателями основ сообщества Linux; другие предлагают «живые» IRC-каналы или шоу, посвященные Linux. Третьи, пользуясь дешевой вещанием через Internet, предоставляют загрузку видео. Итак, недолго думая, настроимся и поищем, что имеет смысл пристегнуть к вашему списку воспроизведения.

«Если в 2007-м создать подкаст было просто, то сейчас это элементарно.»

The Linux Link Tech Show

У семи нянек?..

Ведите слова 'Linux' и 'podcast' в любом поисковике – и прямоком угодите в это шоу, доступное в форматах MP3, OGG и SPEEX.

Каждую неделю, начиная со своего дебюта в 2003 году, команда потоком выпускает эпизоды, и первое, что вас поразит – их длительность.

Выпуски зашкаливают за два часа; само по себе это вроде и не криминал, кабы затягивание не вызывалось отсутствием четкого сценария и обилием посторонней болтовни.

Лучше меньше, да лучше

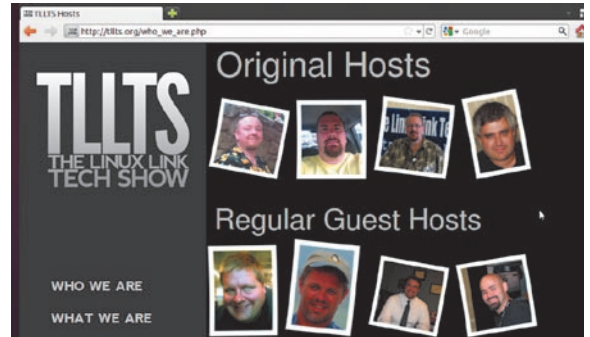
Еще одна особенность – количество людей в студии, в среднем более полудюжины. Множество высококвалифицированных ораторов – это плюс, но только пока они не принимают говорить все сразу, что бывает отнюдь не редко. Эпизоды имеют заглавия, но эти заглавия отражают лишь тему, разговоры на которую ведутся дольше всего.

Ведущие стараются вовлечь в прямой эфир сообщество Linux – через свой IRC-канал, а также блоги и форумы. Фактически, заметки к шоу представляют собой журналы IRC, захваченные во время вещания.

Приглашенные гости

Время от времени на шоу присутствуют гости. В эпизоде 400 появлялся Фабиан Шершель из Linux Outlaws и рассказывал о проекте OGG Camp. Шоу завершается выбранной «песней недели», и в указанном эпизоде фигурировала песня *Dropping out of School* [Слинять из школы] Брэда Сакса [Brad Sucks] – между прочим, принятая в качестве музыкальной заставки TuxRadar.

«Затягивание вызывается обилием посторонней болтовни.»



» Если бы эти парни писали сценарии своего шоу чуть лучше, большое бы им спасибо.

LINUX FORMAT **Вердикт**

The Linux Link Tech Show

Сайт: <http://tlts.org>

» Качественная информация, но многовато болтовни, и эпизоды затянуты.

Рейтинг 6/10

Linux Outlaws

Двое разбойников.

Единственное, что есть в этом шоу слегка ненормативного [англ. out-law – изгой, человек вне закона] – это лексика. Но чего и ожидать от шоу, о котором часто говорят, что оно подхватило знамя популярного, но ныне закрывшегося LUG Radio.

Шоу собрало значительную аудиторию с момента его запуска в сентябре 2007, и они заявляют, что достигли уровня более 50 000 загрузок в месяц.

Народное голосование

Ведущие Фабиан Шершель [Fabian Scherschel] и Дэн Линч [Dan Lynch] заработали немалую популярность в сообществе подкастинга и часто фигурируют в других подкастах как гости или соведущие. Хотя они географически удалены друг от друга, качество аудио в этом шоу безупречное. Впрочем, оно и неудивительно: один из ведущих – заядлый меломан.

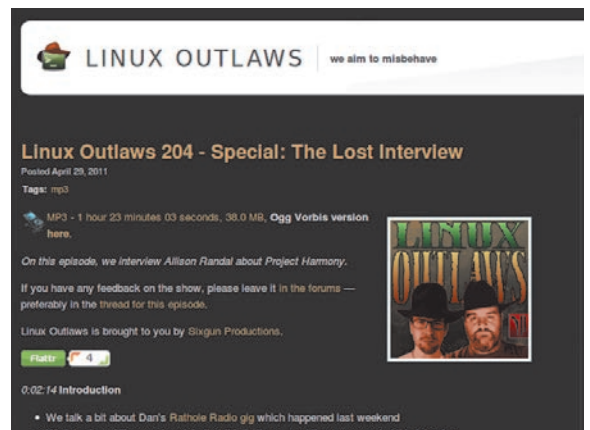
Шоу выходит еженедельно в форматах MP3 и OGG. Длительность варьируется: от 40 минут, если неделя бедна новостями, до двух часов – если ведущие рассматривают актуальное ПО или берут интер-

вью у известных членов сообщества Open Source. Новости подаются квалифицированно, хотя и в несколько категоричной форме, но главная изюминка шоу – интервью. Оба ведущих отлично информированы о делах сообщества и прекрасно дополняют друг друга.

В последнем шоу было подробное интервью с Эллисон Рэндел [Allison Randal, архитектор Ubuntu], которое началось с дискуссии об identi.ca. После интервью ведущие обсудили с Рэндел проект Harmony и вопросы передачи авторских прав. Вдобавок они побеседовали с Рэндел и о других проектах, в частности, Python 3.

В регулярных эпизодах они рассматривают актуальные релизы, а завершают обратной связью с аудиторией. Эта команда отлично справляется с вовлечением аудитории через форумы, блоги, IRC и собственную службу новостей.

«Главная изюминка шоу – интервью с лидерами сообщества.»



» По коням! Ребята зовут вас учинить наезд!

LINUX FORMAT **Вердикт**

Linux Outlaws

Сайт: www.linuxoutlaws.com

» Отлично спланированное юморное шоу; слушать его – сплошное удовольствие.

Рейтинг 9/10

Going Linux

Идет раз... идет два... продано!

Подкаст для новичков... звучит ерундой, правда? Мы были настроены скептически, но, к нашему изумлению, этот подкаст – всерьез и надолго. Вместо регулярной пробежки по новостям, ведущие Going Linux ориентируются на аудиторию из новичков или людей, планирующих перейти на Linux, и растолковывают всю подоплеку каждой новости.

Три за раз

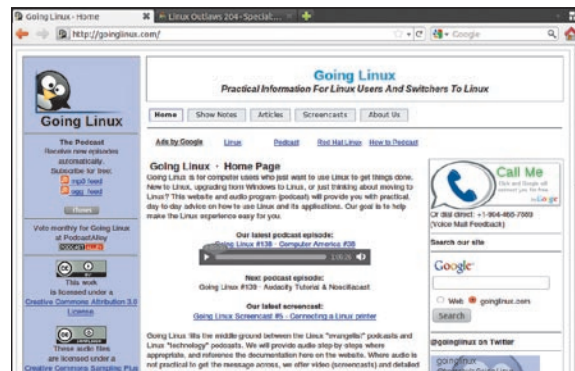
Шоу длится 60 минут и доступно для загрузки в форматах MP3 и OGG. Оно выходит раз в два месяца, но его ведущий Ларри Буши [Larry Bushey] ежемесячно приглашается на канал Computer America для бесед о Linux, так что вы, по сути, получаете три выпуска.

Как правило, здесь нет обзоров, за исключением случаев выхода чего-то фундаментального – например, Ubuntu 11.04. Сначала они просто рассказали о своих впечатлениях о Natty и минут 10 поиграли с ним, а потом на Computer America,

в эпизоде 138, Ларри замечательно рассказал о предыстории Unity и Gnome 3, сравнил их с KDE 4.0 и дал введение в эти среды для пользователей Windows и Mac.

Нам очень понравилось, что, сославшись на Audible.com как на спонсора шоу, ведущие упомянули, что у сервиса до сих пор нет Linux-клиента, и предложили слушателям писать в компанию запросы на его разработку. Значительная часть шоу посвящена помощи пользователям в решении их задач с помощью приложений Linux, рекомендациям полезных обучающих материалов и ответам на вопросы.

Быть может, в зависимости от вашего уровня подготовки вам понадобятся не все эпизоды, но они весьма подробно расписаны по времени. Проверьте: вдруг будет обсуждаться интересное вам приложение.



➤ На сайте есть набор нетленки для начинающих пользователей Linux.

LINUX Вердикт
FORMAT

Going Linux

Сайт: www.goinglinux.com

» Хорошо продуманное шоу для новичков в Linux.

Рейтинг 7/10

«Ведущие растолковывают всю подоплеку каждой новости.»

Linux Basement

Байки из склепа.

Последний эпизод этого подкаста назывался «Пока еще живы». Обычно эта команда выпускает по паре эпизодов в месяц, но между последними был разрыв в два месяца. Тем не менее, несмотря на нерегулярность выходов, это один из лучших подкастов в обзоре.

Хотя основатель Чед Уоллонберг [Chad Wollonberg] и соведущий Клаудио Миранда [Claudio Miranda] изо всех сил стараются создавать эпизоды, уместные для семейного просмотра, они отвели душу в разнузданной критике Canonical, выйдя в эфир с разговором о *Banshee*.

Флеймы

Подкасты тенденциозны уже по своей природе. Хотя многие из аргументов, приведенных против Canonical, были оправданны, многие слушатели обиделись и забросали ведущих разгневанными письмами.

В последнем эпизоде, который мы слушали и который не планировалось вы-

пустить до South East Linux Fest (SELF), ведущие отлично справились с обзором Ubuntu Natty Narwhal.

Подкаст доступен в форматах MP3 и OGG; качество звука обычно очень хорошее, но в последней трансляции время от времени наблюдались помехи.

Простой формат

Формат шоу Linux Basement [англ. подвал] прост: сперва – обсуждение важных новостей, затем следует необычный обзор. Иногда ведущие приглашают гостей, а их заметки представляют собой набор ссылок на новости, обсуждавшиеся в каждом эпизоде.

Каждый эпизод продолжается от 70 до 90 минут, и для подкаста именно такое время и оптимально.

«Они отвели душу в разнузданной критике Canonical.»



➤ Не все тут может понравиться, но мы неравнодушны к шоу с двумя ведущими.

LINUX Вердикт
FORMAT

Linux Basement

Сайт: www.linuxbasement.com

» Стоит послушать, если в заметках к шоу упоминается нечто интересное вам.

Рейтинг 6/10

The Linux Action Show

По методике Панча и Джуди.

Это еженедельное шоу, один из наиболее занимательных подкастов Linux, пару раз меняло имена, и продолжается уже 16-й сезон. Выпускает его Jupiter Broadcasting; шоу доступно в форматах OGG и MP3, а также как видео MP4 в нескольких разрешениях для разных устройств.

Продолжительность – от 40 до 75 минут, ведущие – Брайан Ландьюк [Bryan Lunduke] из Illumination Software Creator и Крис Фишер [Chris Fisher], основатель Uper Broadcasting.

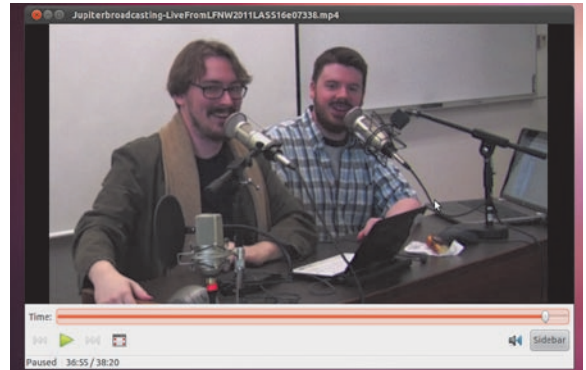
Звонок к началу

После оригинальной звуковой заставки ведущие представляют приложение недели – для Android и для настольного компьютера. Затем они обсуждают новости, и, по сравнению с другими подкастами, их комментарии довольно безапелляционны. Гротескная театральность иногда вредит восприятию. Говорят ведущие четко, но настолько быстро, что не-американцам трудно их понимать.

Трансляция обычно идет из студии, но когда мы занимались составлением данного сравнения, ведущие посетили конференцию LinuxFest Northwest (LFWN) и пригласили ведущего FLOSS Weekly, Рэндела Шварца, обсудить новости. Затем последовало интервью с Аланом Кларком [Alan Clark] из OpenSUSE и Linux Foundation.

В следующем эпизоде с LFWN они осмеивали интерфейс Ubuntu 11.04 Unity. Баланса ради, они пригласили Джоно Бэкона [Jono Bacon] поговорить о перспективах Ubuntu – к сожалению, качество звука было не лучшим. Это один из немногих подкастов Linux, записываемых в контролируемой среде, и качество его обычно на высоте, хотя слова и изображения иногда рассинхронизированы.

«Гротескная театральность иногда вредит восприятию.»



» В видеовersion подкаста вы увидите, как Брайан мурлычет свое фирменное вступление.

LINUX FORMAT **Вердикт**

The Linux Action Show

Сайт: www.jupiterbroadcasting.com

» Занимательный подкаст с эксцентричным юмором.

Рейтинг 8/10

MintCast

Легкий, свежий и живительный.

Это шоу выдержано в том же русле, что и предыдущее, и производится членами сообщества Linux Mint. В отличие от подкаста Full Circle, это шоу идет чуть дальше и затрагивает новости и других дистрибутивов. Оно также включает дискуссию за круглым столом – обычно в ней участвует четверо и более ведущих, выступающих с разной степенью активности.

Рекомендуем MintCast всем пользователям Mint. Этот подкаст выходит еженедельно, длится от 60 до 90 минут. Он доступен в форматах MP3 и OGG, качество звука очень высокое, а для шоу с большим количеством участников это редкость.

Смешанная картина

В начале эпизодов каждый ведущий выступает с кратким введением, а за этим следует сводка новостей. Наряду с новостями о различных выпусках Mint, они касаются и других тем: например, релиз GIMP, противоречия, окружающие Dropbox, и выход

Fedora 15. В последних эпизодах лучшими моментами были обзоры релиза Mint Xfce, Ubuntu 11.04 и приемов программирования в Linux Mint.

В ходе шоу выбирается сайт недели – обычно он связан с основной темой эпизода, например, советом недели. Отведено и время на обратную связь со слушателями. Ведущие не жалеют усилий на составление подробных заметок к шоу с привязкой по времени, что служит для него прекрасным дополнением.

Этот подкаст не претендует на роль «универмага» всех новостей мира Linux, но он оценивает все важное и описывает, как это может повлиять на экосистему дистрибутивов Linux. Нам нравится, что ведущие не кичатся своими знаниями в области Linux, а потому слушать их приятно.

«Наряду с новостями о выпусках Mint они касаются и других тем.»



» Эти ребята не только слушают свою аудиторию – одного они даже пригласили на шоу!

LINUX FORMAT **Вердикт**

MintCast

Сайт: <http://www.mintcast.org>

» Дружелюбный к новичкам подкаст с упором на Linux Mint.

Рейтинг 8/10

FLOSS Weekly

Ваша недельная доза бесед.

Еще один подкаст с вывертом. FLOSS не выдает регулярной порции новостей и обзоров: он сосредоточен на интервью, причем очень подробных.

Это шоу – часть сети TWiT, которая надеется стать CNN для компьютерных фанатов. Как следует из названия, подкаст еженедельный; он доступен в формате MP3, а также как видеоролики MP4 с парой разных разрешений. Версии OGG нет, и использование закрытого ПО не добавляет шоу новых приверженцев. В одном из эпизодов ведущий Рэндел Шварц [Randal Schwartz] даже признал, что довольно неловко приглашать гостей на шоу об открытом ПО, пользуясь Skype.

Вращающаяся панель

Партнеры Шварца отображаются на вращающейся панели с портретами соведущих, и в каждом эпизоде он приглашает гостей для обсуждения проектов Open Source, а те определяют, кто будет со-

ведущим в этот раз. Например, на интервью с разработчиками *OpenCSW*, пакетного менеджера для Solaris, соведущим был Саймон Фиппс [Simon Phipps], ранее работавший в Sun Microsystems.

Вопреки сложившейся традиции, на последнем шоу, со зловещим названием *The Demise of FLOSS* [Гибель FLOSS], гостя не было, и ведущие обсуждали новости, в частности, отлынивание разработчиков Mono и то, как это скажется на проектах типа *Banshee*.

Эпизоды длятся примерно час, и проблем с качеством аудио и видео нет – благодаря корпорации, стоящей за спиной шоу; но нет и заметок к шоу, достойных упоминания.



► Прибавив щепотку новостей, шоу могло бы стать лучшим подкастом о Linux.

LINUX FORMAT **Вердикт**

FLOSS Weekly

Сайт: <http://twit.tv/FLOSS>

» Подкаст удобный, интервью подробные, но почти ничего сверх этого.

Рейтинг 7/10

«Использование закрытого ПО не добавляет приверженцев.»

Hacker Public Radio

Непосредственное и безыскусное.

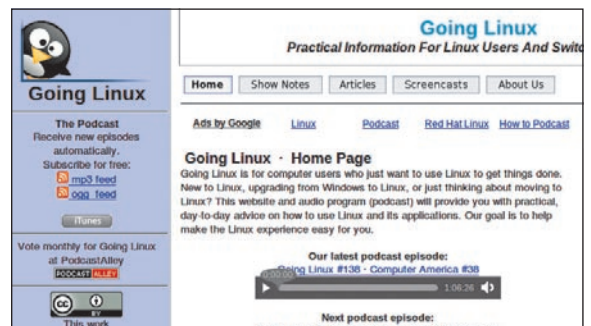
Шоу весьма интересное. Хотя декларируется, что оно посвящено «всему, что интересно хакера», основная его направленность – Linux. Здесь нет постоянных ведущих, но шоу уникально тем, что поощряет слушателей из Великобритании и США внести свой вклад «звонками на студию».

Горести Windows

Популярная тема – как слушатели перешли на Linux и почему отказались от Windows. Постоянные соведущие вносят в каждый эпизод различные темы – например, как обеспечить безопасность компьютера или как взломать телефон Android.

Вы найдете там фрагменты, внесенные командами других подкастов, например GNU World Order и Full Circle. Постоянные ведущие ежемесячно собираются и делают обзоры самых интересных шоу за прошлый месяц.

Длина эпизодов зависит от их типа, как и качество информации и аудио. Это не дискриминация – но не каждый выпуск хочется послушать.



► Набралось уже более 1000 эпизодов этого ежедневного шоу.

LINUX FORMAT **Вердикт**

Hacker Public Radio

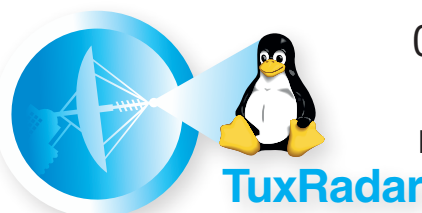
Сайт: www.hackerpublicradio.org

» Будьте разборчивы, и непременно натолкнетесь на что-то полезное.

Рейтинг 5/10

«Шоу уникально тем, что поощряет вклад «звонками в студию».»

Не пропустите наш подкаст: www.tuxradar.com



Слушайте, как команда Linux Format навещает горячие точки планеты свободного ПО!

Подкасты Linux

Вердикт

Если вы слушали хоть одно такое шоу, поверьте, что выбрать победителя было нелегко, потому что все они нацелены на собственную аудиторию. Новичкам в мире Linux мы рекомендуем Going Linux, хотя MintCast и Full Circle тоже подойдут пользователям дистрибутивов для начинающих.

Линуксоиды-ветераны сочтут полезным время от времени ловить волну Hacker Public Radio. Если вы ищете забавное и информативное шоу, чтобы держать руку на пульсе экосистемы Linux, получать новости от информированных ведущих и слушать интервью с лидерами сообщества Open Source, то не найдете лучше, чем продукт «ковбоев микшера» Linux Outlaws. В нашем сравнении от 2007 года, при всего пяти выходах они были сала-

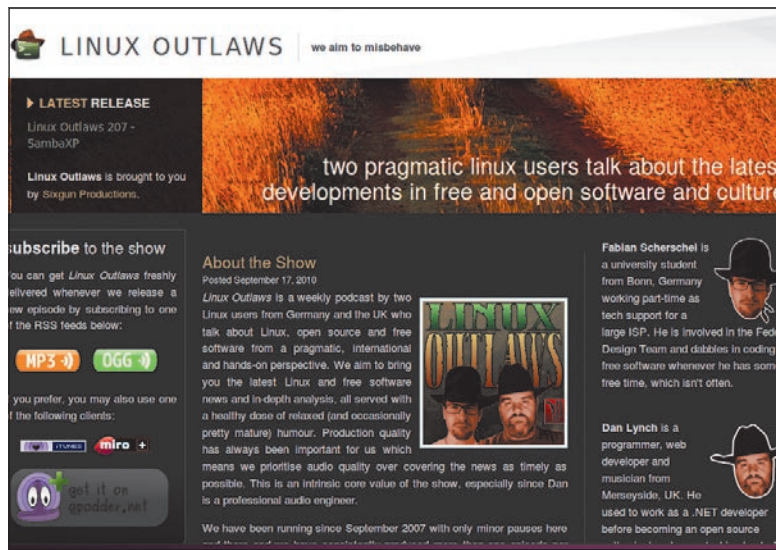
гами, но теперь эпизодов в их активе более 200, и этот еженедельник привлек соратников во всем мире.

Все существующие подкасты Linux здесь не пересравниваешь. Если вам не нравится наш вердикт, знайте: выбор го-

«Не нравится наш вердикт? Выбор гораздо шире.»

раздо шире. Опытные пользователи Linux могут склониться в пользу Kernel Panic, the Techie Geek, Just Talk Geek to Me или таких шоу без ведущих, как GNU World Order и Lotta Linux Links.

Летунам с дистрибутива на дистрибутив следует слушать DistroWatch Weekly



I Linux Outlaws ★★★★★
 Сайт: www.linuxoutlaws.com
 » Популярное и хорошо информированные ведущие, много полезных Linux-новостей.

II Linux Action Show ★★★★★
 Сайт: www.jupiterbroadcasting.com
 » Весьма увлекательно, и тоже много полезных новостей о Linux.

III MintCast ★★★
 Сайт: www.mintcast.org
 » Бесценно для пользователей Mint, но и всем прочим пригодится.

IV FLOSS Weekly ★★★
 Сайт: <http://twittv/FLOSS>
 » Гости недели помогают создавать информативное и интересное шоу.

и Ubuntu-UK. Если вас интересует свободное ПО, обратите внимание на подкаст Software Freedom Law Show, который ведет Брэдли Кун [Bradley Kuhn], и шоу Free Software Round Table. Есть и шоу, основное внимание уделяющие программному обеспечению для Linux – например, SourceTrunk, Linux Basix, которые содержат советы и рекомендации для новичков, и, наконец, Linux For the Rest of Us, которое заодно транслирует и видео.

Говоря о видео – стоит обратить внимание на подкасты The Source Show и This Week in Linux. **LXF**

» **Пожизненный запас истинно британских чипсов для желающих посмотреть «макаронный вестерн» подкаста о Linux, который ведут два европейца.**

Изучите цифры

Название	Частота выхода	Количество эпизодов	Огг	MP3	Заметки к шоу	Длительность
Linux Link Tech Show	Еженедельно	401	Да	Да	Нет	2:03
Linux Basement	Раз в два месяца	69	Да	Да	Да	1:50
Hacker Public Radio	Ежедневно	727	Да	Да	Нет	–
Linux Outlaws	Еженедельно	204	Да	Да	Да	1:23
The Linux Action Show	Еженедельно	168	Да	Да	Да	0:51
Going Linux	Раз в два месяца	138	Да	Да	Да	1:06
FLOSS Weekly	Еженедельно	165	Нет	Да	Нет	0:50
MintCast	Еженедельно	64	Да	Да	Да	1:24



Матч дистрибутивов

Мы выбрали самые популярные дистрибутивы дня и устроили гладиаторскую битву за ваши симпатии. Исчерпывающие испытания и скороспелые выводы, ау!

В начале Линус [Linus] создал ядро. Ядро работало (более-менее), и было хорошо. Затем, от растущего взрыва кода, мир получил тысячи различных дистрибутивов Linux — некоторые, кажется, отличаются друг от друга только цветом обоев рабочего стола.

Выбор дистрибутива — непростая задача, она занимает много времени и требует больших усилий, и поэтому пользователи Linux не отходят далеко от обновлений того, что сами лучше знают.

Такая лень достойна одобрения, но поскольку дистрибутивы конкурируют друг с другом на разных фронтах, вы, возможно, теряете то, что больше вам подошло бы. Вам интересны самые свежие свободные програм-

мы, или вас больше волнует безопасность? Что вам нравится больше, рабочий стол Gnome или KDE? Хотите ли вы, чтобы дистрибутив сам выбирал для вас программы, или лелеете перспективу собственноручного выбора каждого пакета?

Ответы на эти вопросы, и еще на тысячи сопутствующих им, приняли форму сравнения на следующих семи страницах. Сопоставить можно много позиций, и на чашу весов

брошено много дистрибутивов — захоти вы сравнить их сами, вам потребовалось бы две недели головной боли, с короткими перерывами на сон, чтобы оценить результаты изме-

рений. Но не парьтесь: готовые результаты у нас уже есть. С диаграммами и графиками.

«Вы, возможно, теряете то, что больше вам подошло бы.»

Установка

Первая препона для дистрибутива – дать пользователю им... воспользоваться.

В отличие от Windows и Mac OS, пользователям приходится устанавливать ОС с нуля, что, конечно, является барьером для перехода. В ранние годы установщиков пользователям приходилось знать много технической информации об их системах, но сейчас, к счастью, даже самый примитивный дистрибутив прост в установке.

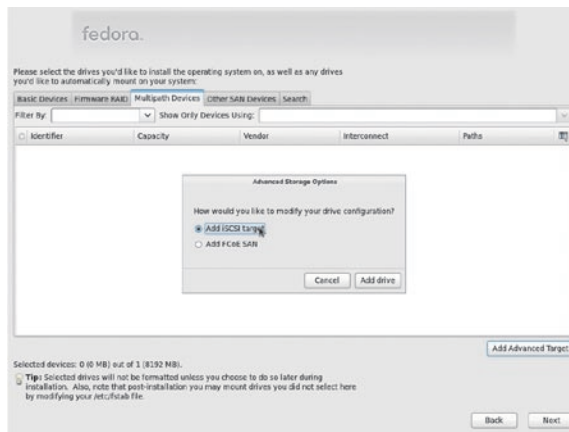
Тем не менее, текстовый интерфейс программы установки, применяемый в Arch Linux и Debian, безусловно озаботит некоторых пользователей. Установщик задает все те же вопросы, однако обоим дистрибутивам есть над чем поработать по части разбивки жесткого диска. Даже опытного пользователя вгонит в ступор попадание в командную строку во время приготовления жесткого диска. Графический установщик в Debian есть (хотя по умолчанию используется текстовый); он, конечно, задает много вопросов, но управлять им несложно. Однако не пытайтесь пользоваться им, не имея сетевого соединения.

Mint и Ubuntu, естественно, очень похожи: оба используют установщик Ubiquity, хотя и немного по-разному. Тот спроектирован так, чтобы запускаться из работающей ОС, поэтому при установке оба полагаются на live-дистрибутив.

В Ubuntu это не так заметно, так как при загрузке он выдает меню, где можно выбрать между запуском live-версии и самой установкой.

За последние годы Ubiquity сильно изменился, и определенно стал более дружелюбным и надежным, чем раньше. Он автоматически разбивает за вас жесткий диск и обходится минимумом вопросов, одновременно копируя файлы. Mint, естественно, отличается только цветовой схемой.

Установщик OpenSUSE окрашен в спокойный зеленый цвет, и успокоительное тут действительно пригодится. На первый




взгляд он ошарашивает сложностью, так как почти что «вместо здравствуйте» предлагает указать разнообразные опции для режимов экрана и параметров ядра. Но на самом деле это предотвращает проблемы с нетипичной графической картой или устройством хранения, которые вы не смогли бы решить.


Установщик столь же дружелюбен, но более подробен. Как и в Fedora, по умолчанию метод установки – с DVD, с набором программ, которые можно тут же установить. Недостаток такого подхода в том, что если вам неохота тратить долгие часы на отбор пакетов, то почти наверняка у вас появятся пакеты из групп, которые вы никогда не используете и даже не знаете, что они есть.


По времени установки быстрее всех Debian и Arch, но здесь тест не совсем справедлив, так как оба, и особенно Arch, ставят только самый костяк.

» У Fedora установка по умолчанию делается с DVD.

ПОДИУМ

 **FEDORA**

 **UBUNTU**

 **OPENSUSE**

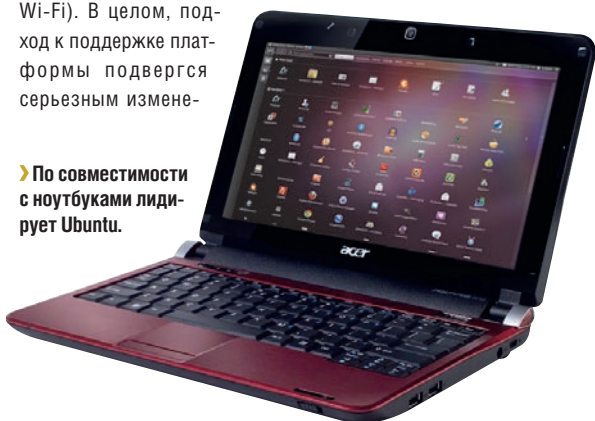
Вердикт
» Побеждает Fedora – за простоту, надежность и дружелюбие.

Поддержка оборудования

Ладит ли ваш дистрибутив с платформами и периферией?

Способов определения поддержки оборудования несколько. Два главных – это поддержка платформы (процессор/материнская плата/монитор и так далее) и периферии (принтер/web-камера/Wi-Fi). В целом, подход к поддержке платформы подвергся серьезным измене-

» По совместимости с ноутбуками лидирует Ubuntu.



ниям. Если раньше дистрибутивам приходилось предоставлять множество драйверов третьих лиц, то сейчас команда ядра многое взяла на себя – то, что раньше было проблемой, сейчас просто очередной модуль в ядре.

С точки зрения платформы, очень сложно опередить Debian, главным образом потому, что это чуть ли не единственный дистрибутив, пригодный для установки на Power PC (например, старых Mac), мейнфреймах S390 и всем, что между ними.

Трудностью для многих дистрибутивов остается область ноутбуков, где компоненты замены нередко не подлежат и ощутимо отличаются от аналогов для настольных машин.

В этой области лидирует Ubuntu, в основном благодаря своей популярности – многие производители, снизошедшие до Linux (Dell, Acer и пр.), более или менее придерживаются стандарта Ubuntu, так что у него (и его представителя Mint), возможно, самые высокие шансы запуститься на произвольном ноутбуке.

Победа в этой номинации присуждается Ubuntu и SUSE – хотя бы за старания сохранить список совместимости; например, Fedora давным-давно плюнул на эти попытки.

ПОДИУМ

 **UBUNTU**

 **MINT**

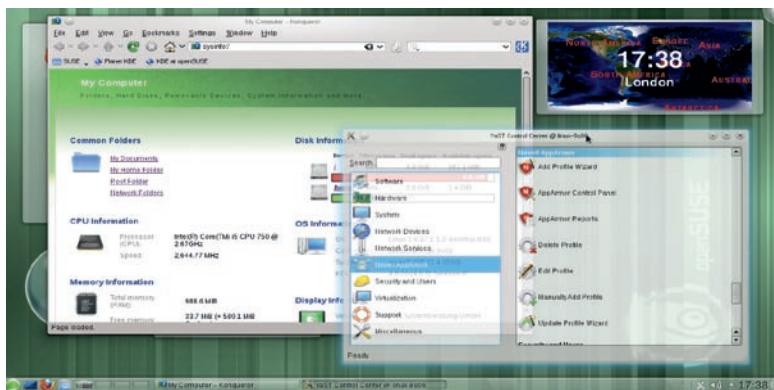
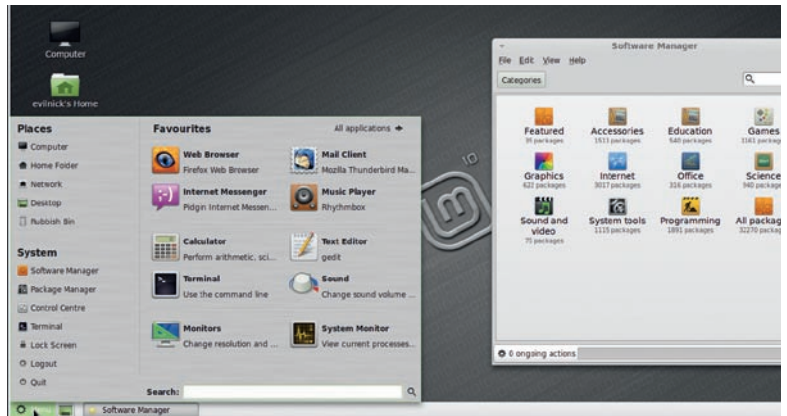
 **DEBIAN**

Вердикт
» Побеждает Ubuntu, благодаря совместимости с ноутбуками.

Linux Mint

Этот прямой и непосредственный рабочий стол может стать тихой гаванью для потерпевших от бурных изменений Unity в Ubuntu или Gnome 3 в Fedora или где-нибудь еще. Mint уже стал очень популярным ответвлением Ubuntu, и может ощутить приток новых рекрутов. Хотя Mint основан на Ubuntu, следующий его релиз (11, «Катя») не будет воспроизводить переход на Unity, и таких планов на будущее не строят: сейчас их цель – создать лучший рабочий стол Gnome 2.

Взлету этот дистрибутив обязан близости к Ubuntu при одновременном наличии той проприетарной начинки (*Flash, Java*, графические драйверы), которую в других дистрибутивах приходится добывать после установки. Но сейчас в Mint появилось много чего другого, и он во многом проще и дружелюбнее Ubuntu.



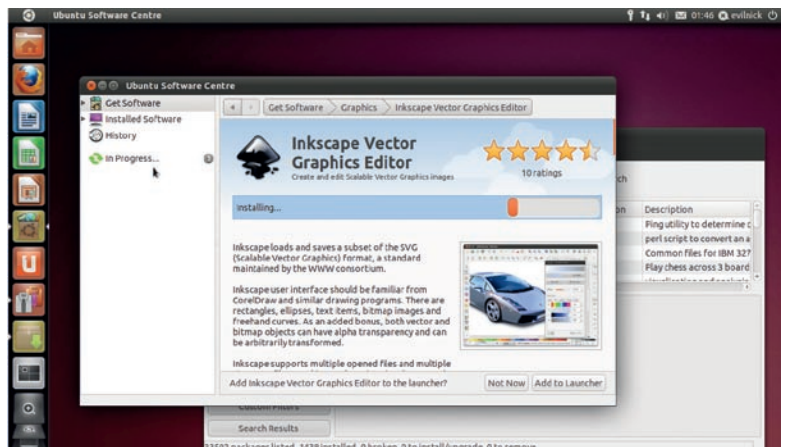
OpenSUSE

OpenSUSE – единственный дистрибутив, выбравший KDE в качестве рабочего стола по умолчанию, хотя с тем же успехом уживается с Gnome 2 (гораздо лучше, чем, например, Fedora с KDE). Хотя победителем без соперников стать легко, невольно уважаешь кураторов пакетов за такую приверженность ко KDE.

Если вы пробовали KDE и он вам не понравился, то, возможно, это было в дистрибутиве, который не уделяет ему такой заботы, как OpenSUSE. Ошибиться очень сложно, хотя при установке других программ и приходится ладить с монструозным *Yast*. Приложения KDE несколько не уступают своим аналогам из Gnome, хотя вместо *Amarok*, конечно же, можно установить *Rythmbox*, и т. п.

Ubuntu

Долгое время единственным игроком в Ubuntu был Gnome, но сейчас все изменилось. Рабочим столом по умолчанию стал Unity [англ. единство] – элегантно-минималистский, с широкой боковой панелью, служащей и для запуска популярных приложений, и для переключения между рабочими полями. На него уже поступили отрицательные отзывы, которые не объяснить обычной боязнью перемен. В Unity существуют проблемы, выходящие за рамки простого неприятия его пользователями, и, несомненно, разработчикам придется крепко поработать над ошибками. Есть проблемы и с функциональностью. Как изменить размер дока? Где виджет системного монитора?.. Конечно, можно выбрать запуск в «стандартном» Gnome, однако возможности выйти в Gnome 3 нет. Похоже, именно «единства» и недостает рабочему столу Ubuntu, по крайней мере, в настоящий момент.



Настройка

Свобода как механизм выбора – в этом весь Linux.

Приближенная мера гибкости настройки вашей системы – количество доступных в ней рабочих столов, тех, что задаются при установке. Здесь менее жестко структурированные проекты, Arch и Debian, стоят особняком – хоть и с лишними усилиями, но в них можно установить практически любой рабочий стол без потери функциональности.

Некоторые дистрибутивы заботятся о других опциях, но, например, в Fedora все административные утилиты заточены под Gnome, и здравая интеграция системы может оказаться довольно сложной. Ubuntu вообще не имеет официальной под-

держки рабочих столов на базе KDE, тогда как OpenSUSE ладит и с Gnome, и с рядом других.

Если Arch и Debian являются более гибкими (в конце концов, их легко настроить на чисто текстовую установку), то на сцену вступает другой фактор: собственно инструменты настройки. Большие дистрибутивы предоставляют самый широкий выбор пакетов; Ubuntu и Linux Mint стягивают пакеты из гегантского репозитория Debian, где только свободных пакетов более 28000, без учета несвободных и зависящих от несвободных источников.



Сообщество

Сбор измерений по использованию Linux сложнее, чем кажется.

Каким способом это ни измеришь, всегда будет существенная неопределенность в достоверности данных, вплоть до их бесполезности. Что брать в расчет – число скачиваний? Зарегистрированных пользователей на Linux Counter или на главном форуме? Постов по теме в LinuxQuestions? Все эти показатели дадут разные результаты – причем до ужаса разные.

При любом выбранном показателе, ясно будет одно: большая часть трафика указывает на Ubuntu; в плане пользовательской базы он затмевает всех остальных.

Стоит ли удивляться, если учитывать, что Ubuntu – практически единственная версия Linux, которую можно найти предустановленной на чем угодно.

Ubuntu также занимает видное место в прессе – большинство основных компьютерных изданий полагает, что Ubuntu и Linux – синонимы, и неудивительно, что в этой мысли утвердились и пользователи.

Но отражается ли «сообщество» простым количеством пользователей? Можно привести убедительные доводы, что пользователи Debian, Fedora и OpenSUSE во многих областях активнее, включая вклад в код, документацию и справку. Цифры из реального мира подтверждают это.

На форумах дистрибутивов LinuxQuestions.org большинство постов расположено в разделе Debian, за ним по пятам следует Fedora, а Ubuntu томится на четвертом месте, после SUSE.

Это не говорит о том, что посетители сайта более склонны к запуску Debian, чем Ubuntu, или, по крайней мере, к разговору о нем, но все же весьма показательно, коль скоро Ubuntu считается несомненным лидером рынка.



➤ Анализ новостей и поисков в Google выдает, что Ubuntu гораздо больше на слуху, чем другие.

Судя по анализу тенденций web-поиска или новостей о дистрибутивах от Google, можно заключить, что Ubuntu используется больше, чем все остальные дистрибутивы, вместе взятые.

Какие выводы? С точки зрения пользователя рабочего стола все указывает на Ubuntu, а остальные далеко позади. С точки зрения активных членов сообщества, все выглядит так, будто большая часть этих пользователей Ubuntu – просто пользователи, не принимающие активного участия в «сообществе». Это неудивительно, ведь Ubuntu – это путь наименьшего сопротивления, от охвата в СМИ до доступности. Если определять сообщество как долю активных пользователей, менее привычные дистрибутивы имеют гораздо лучшие показатели.

ПОДИУМ

DEBIAN (Gold medal)

UBUNTU (Silver medal)

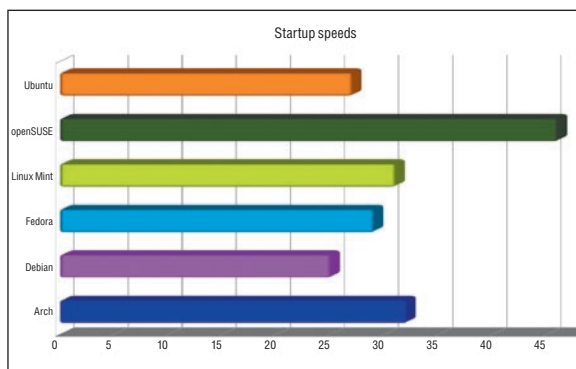
FEDORA (Bronze medal)

Вердикт
 ➤ Сообщество Linux нельзя выразить голыми цифрами.

Производительность

Сейчас мы делаем такое, о чем раньше и не мечтали.

Выжимание каждого свободного такта CPU уже не кажется сейчас чем-то важным. Феноменально большая тактовая частота и многоядерность современных процессоров приводят к тому, что 90 % времени они не работают на полную катушку (в Башнях LXF даже больше 90 %).



➤ Видно, что при запуске всех обогнал OpenSUSE.

Однако независимо от скорости компьютеров нам все равно мерещится, что нечеловечески много времени проходит в ожидании, когда они выполнят задачу, и это вроде бы немного нелогично. Кто-нибудь из читателей работал за Amiga? Его тактовая частота была 7 МГц. Сегодня у среднего CPU она около 3 ГГц, и редкий процессор не является многоядерным, то есть современный компьютер быстрее в 800 раз! Тем не менее загрузка выглядит долгой.

По части скорости работы приложений между дистрибутивами большой разницы нет. В одних, таких как Fedora, ядро очень хорошо настроено, но ускорение можно наблюдать только в больших масштабах. Другие, наподобие Arch, тонкие и легкие, то есть если не перегружать их программами, они будут работать быстрее.

Ключевое различие – в скорости запуска, и по ней мы составили таблицу. Мы выполнили несколько других простых тестов, но разница всех времен укладывалась в пределы погрешности.

С солдатской прямоотой скажем: выбор рабочего стола, графического драйвера и объем ОЗУ играют гораздо большую роль для ускорения работы, чем выбор дистрибутива, если только вы не создаете высокопроизводительный кластер.

ПОДИУМ

DEBIAN (Gold medal)

ARCH (Silver medal)

FEDORA (Bronze medal)

Вердикт
 ➤ Решение было непростым, но Debian протиснулся в лидеры.

Управление пакетами

Ну есть ли на свете более колючая тема?

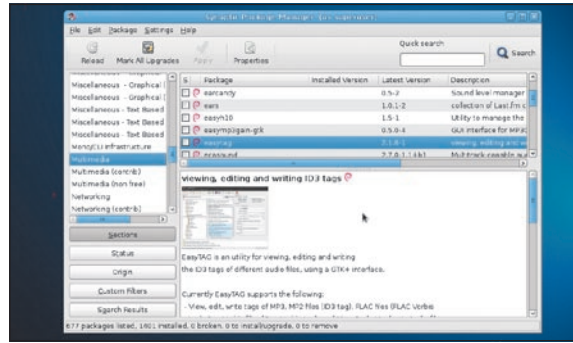
Первое, что вам нужно знать про управление пакетами: согласие, каким оно должно быть, не достигнуто. Когда-то дискуссия Deb (формат пакета Debian) против RPM (система управления пакетами Red Hat) была столь же бурной, как KDE против Gnome, особенно после того, как RPM сделали официальным форматом Linux Standards Base.

Arch в этом разделе самый нескладный, поскольку использует собственные пакеты и утилиту командной строки *pacman* для их установки. Как и в Slackware, это не что иное как двоичный файл и скрипт установки, но за их простотой стоит мощная система, где легко добавлять собственные исходники (и где нет пакетов), не путаясь с системой зависимостей. Но следует быть внимательным, поскольку для правильной работы пакетам требуется последующая настройка.

Fedora и OpenSUSE применяют RPM. Управление пакетами в OpenSUSE сейчас производится через *Zypper*, который и работу делает правильно, и в командной строке запускаться умеет.

Подход Fedora к пакетам идет с нескольких направлений. Хотя он полагается на RPM, и вы можете использовать RPM-утилиты, налицо склонность к путанице. Официальный менеджер установки пакетов – *Yum*, он отслеживает зависимости, поддерживает опции и использует разностные дельта-пакеты, чтобы снизить время скачивания обновлений. Раньше его (справедливо) критиковали за медлительность; эта проблема не полностью изжита, но все-таки он работает. Увы, RPM – не такой стандарт, чтобы пакеты для SUSE непременно работали в Fedora, и наоборот.

Остальные три тестируемых дистрибутива используют Debian. При простой, но мощной утилите командной строки и набором




У Fedora установка по умолчанию делается с DVD.

графических интерфейсов это, похоже, в общем удовлетворяет потребности, а богатство доступных пакетов просто поражает. Благодаря нынешней популярности Ubuntu, ситуация, видимо, таковой и останется.


Овации заслужил Linux Mint – за старание скрыть от пользователей ненужные им детали. Установка недостаточно тщательно протестированного ПО наткнется на немало препятствий!

Считаете вы сервис *Launchpad* в Ubuntu предназначенным сугубо для разработчиков или нет, но он предлагает легкий способ установки добавочных пакетов, хотя это может стоить усилий (и нервов!). В том же русле наше признание отдается OpenSUSE за его службу сборки, дающую разработчикам возможность разворачивать пакеты любого типа из исходного кода.

ПОДИУМ

 UBUNTU

 DEBIAN

 OPENSUSE

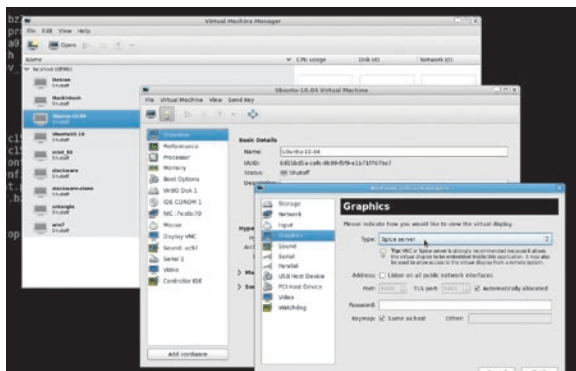
Вердикт

» Благодаря Launchpad Ubuntu обогнал Debian у финиша.

Передний край

Насчет приложений имеется несколько подходов.

Отношение к ПО очень разное: одни дистрибутивы намерены иметь самые свежие программы, другие полагают, что ничего нельзя включать до тщательного тестирования. Третьи пытаются предоставить принятие решения пользователю.



Fedora – по сути, полигон для технологий Red Hat.

Linux Mint очень осторожничает, когда дело доходит до новых программ и обновлений – он решительно не одобряет некритичную установку программ и обновлений, и чтоб хотя бы увидеть список всех доступных пакетов, вам придется влезать в настройки.

Debian верен философии выбора – целый диапазон его репозитория отражает разные уровни риска и награды; беря обновление из «Sid» (устоявшееся прозвище для «нестабильных» релизов), вы должны знать, на что идете. Аналогично, тех, кто включает в Fedora или Arch репозитории «rawhide», ждут веселые денки.

Конечно, существует и несколько уровней ошибок. Вполне возможно, что вам не удастся заставить самую свежую версию работать правильно, или, спутав зависимости, вы сломаете что-нибудь важное. Стоит ли говорить, что за рабочим компьютером такое вытворять не рекомендуется.

Один из девизов Fedora звучит «First [Первый]», и он в самом деле мгновенно реагирует на большую часть технологий – в частности, виртуализацию и хранение данных. Но в этом отношении сложно угнаться за Arch – его расписание скользящих релизов означает, что пакеты доставляются быстро и, большей частью, безопасно. Возможно, они охватывают не все стороны, но это быстрее, чем собирать все самостоятельно.

ПОДИУМ

 ARCH

 FEDORA

 DEBIAN

Вердикт

» Расписание скользящих релизов означает, что Arch недосыгаем.

Безопасность

Обнадеживает, что в этой области есть где порезвиться.

На фронте безопасности есть две новости: хорошая и плохая. Хорошая новость заключается в том, что по части ключевых пакетов все дистрибутивы весьма проворны и обновляют их, как только вредоносный код залатан в исходнике.



➤ Распознав плохой код, все дистрибутивы мигом выдают заплатки, исправления и обновления.

Плохая новость – из-за этого сложно определить, который из них самый безопасный (по крайней мере, при обновленных программах).

В качестве примера возьмем последние предупреждения безопасности в *Apache*, *Asterisk* и, в доказательство, что мы перебрали весь алфавит, *xpdf*.

Все рассмотренные нами уязвимости в основном были залатаны, исправлены и обновлены в один и тот же день – главным образом потому, что поставщики пакетов для разных дистрибутивов тесно связаны с разработчиками программ. Благодаря скользящим релизам, Arch часто обновляет пакеты еще до обнаружения дыр в старых версиях – само по себе это безопасность не повышает, но новые уязвимости надо еще успеть найти.

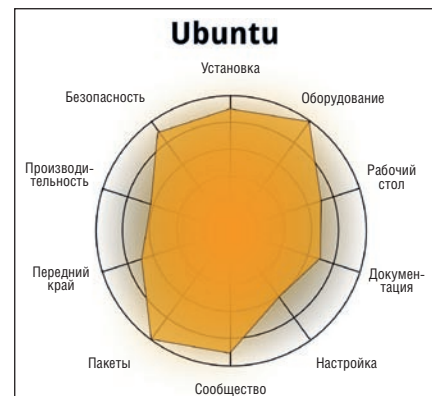
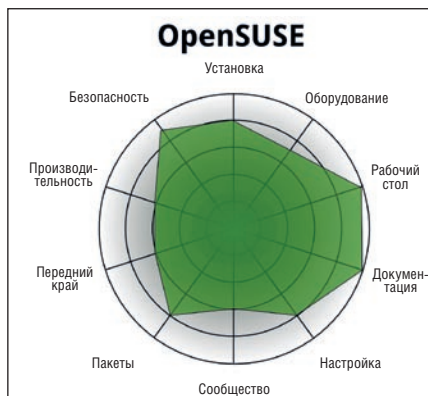
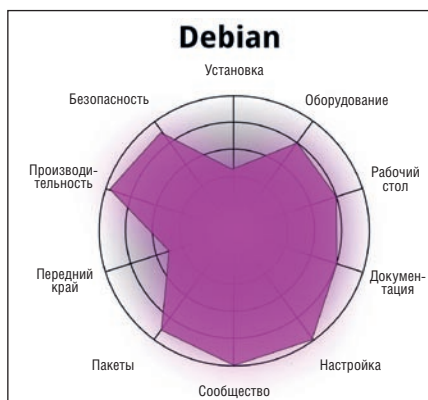
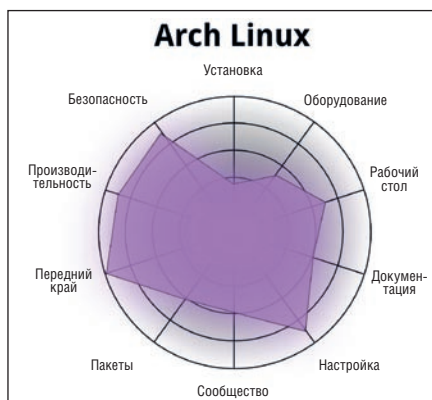
По части мелких проблем безопасности в менее популярном ПО ситуация немного отличается и зависит от интересующих вас пакетов. SUSE и Fedora реагируют очень быстро, и, конечно, при включении обновления в Debian оно также просачивается на Ubuntu и Mint.

Хотя, конечно, большие дистрибутивы и те, что стимулируются бизнесом, как правило, обновляются быстрее, но есть надежда, что при регулярном обновлении вы будете в относительной безопасности независимо от вашей системы.



Статистика

Чего они все стоят...



Вердикт



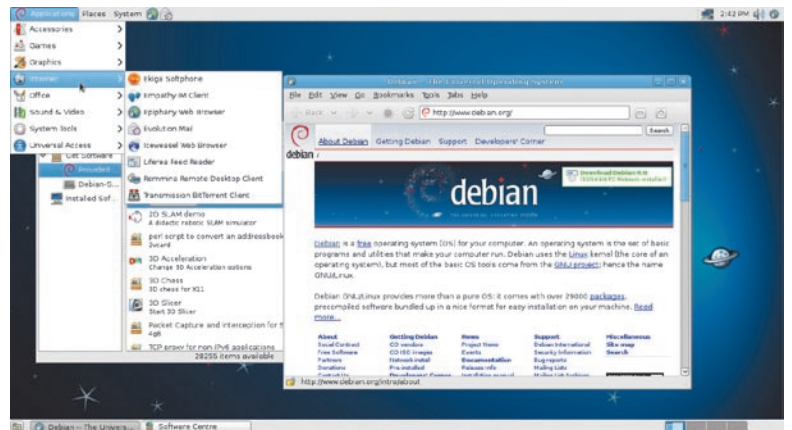
Несмотря на то, что ни одного научного камня на камне не осталось в матрице преобразования, помните, что ни людям, ни пингвинам пока неизвестна наука, способная точно учесть множество параметров, рассмотренных нами в версиях Linux.

Многое абсолютно субъективно и зависит от нужд конкретного пользователя. Начнем с того, что, суммируя таблицу медалей для окончательного результата, мы придали всем категориям равный вес. Честно говоря, это не так – иначе у всех пользователей стоял бы один и тот же дистрибутив.

Однако люди выбирают то, что лучше отражает их потребности. Для некоторых наличие самых свежих программ перевешивает все соображения о сложности их установки, и они предпочитают Arch или Fedora. Некоторым просто нужен простейший способ получить рабочий стол KDE, и они устанавливают OpenSUSE.

У большинства пользователей, скорее всего, стоит Ubuntu, и можно полагать, что предложенный им состав приложений самый правильный. Любопытно, однако, что многие свойства, возвеличившие Ubuntu, обязаны основе в Debian. Также есть фактор Unity. Конечно, отстаивать идею, в которую веришь, очень смело и дерзко, но манера тащить пользователей за уши в рай нового рабочего стола не может не вызвать отторжения, и здесь главные выгоды получает Mint.

Debian – хороший пример дистрибутива на все случаи жизни. В некоторых смыслах он все еще доисторический, и его установка плачет по упрощению – а это позор, ведь многие, испытав трудности на ее этапе, получают дурное впечатление обо всей системе. Debian также чересчур полагается на постоянное сетевое



соединение, что может сделать его стандартную форму неподходящей для нетбуков или установок оффлайн.

Однако его управление пакетами и гибкость – высшего качества, как и многолюдное и активное сообщество, которое предоставляет поддержку, документацию, пакеты, а также широкий спектр мнений. Конечно, подобная деятельность – не для всех, но если вы раньше такого не пробовали, это должно быть самым первым пунктом в вашем списке TO DO.

Если вы гонитесь за простотой в использовании, Linux Mint и Ubuntu – достойные наследники кодовой базы Debian, Fedora и Arch обеспечивают самые новые приложения, а OpenSUSE предоставляет прекрасный разносторонний рабочий стол KDE.

» Debian провозглашает себя универсальной ОС, и по данным наших тестов он прекрасен, как ни крути.

Debian
 » Удивлены? А зря. Прекрасная этика и усилия сообщества приводят к великолепию во всех отношениях.

Ubuntu
 » Могут ли миллионы пользователей ошибаться? О да, но это не относится к живейшей радости от запуска Ubuntu.

Fedora
 » Передний край, с особой аурой; Fedora угадывает лучшее во многих областях.

OpenSUSE
 » Единственно разумный выбор, если нужен современный и профессионально интегрированный рабочий стол KDE.

И прочие...

Конечно, есть и другие дистрибутивы, достойные выбора. Не сердитесь, если здесь отсутствует ваш: мы выбирали просто по текущей популярности. Безусловно достойна упоминания нестесненная яркость Slackware. Он какой угодно, только не slack [англ. расслабившийся], а за его простотой таится мощь незамутненного Linux.

Тем, кто обожает все делать самим, имеет смысл рассмотреть Gentoo и его ответвление Sabayon. А что у нас происходит

с Mandriva? Ну, он все еще существует, за вычетом разработчиков, перешедших в Mageia – см. наш обзор релиза 1.0. Мир Linux изменчив.

И не забудьте про несметное число специализированных дистрибутивов, ориентированных на одну задачу – Jolicloud/JoliOS для нетбуков, Knoppix как хороший live-дистрибутив, CentOS – для бизнесменов, которые не хотят платить Red Hat, и многие, многие, многие другие. LXF

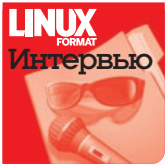


Вим Тейменс

Вим Тейменс

из GStreamer и Collabora

Джоно Бэкон ведет разговор с Вимом Тейменсом, главным разработчиком мультимедиа в GStreamer.



Вим Тейменс [Win Taumans] работает над поддержкой различных кодеков, а также потокового вещания и нелинейного монтажа. Мы поговорили с ним о *GStreamer* и будущем мультимедиа в Linux.

LXF: Как вы впервые втянулись в открытый исходный код?

BT: Должно быть, это было году в 1994 – я тогда установил версию Red Hat Linux на моем Amiga 1200. В итоге я сдался и купил 486-й ПК, но Windows 95 жила у меня дня два, а потом я ее снес и установил Red Hat Linux на постоянной основе. И назад я уже не оглядывался.

Я вошел в проект Utah-GLX – это был веселый способ разузнать об OpenGL и драйверах. Потом к проекту присоединился Джон Кармак [John Carmack] (из *Quake/Doom*) – так я впервые встретился с рок-звездами программирования.

В конце 1998 года я пришел на видеопроект Trinity. Состояние мультимедиа было тогда жалким – практически, среди доступных медиа-плееров самым современным был *Xanim*. Никаких разработок в сфере свободного и открытого кода для современных кодеков не существовало.

Я тогда оптимизировал видеокодек (RTJPEG) для применения в качестве основного формата видео в Trinity, и стало ясно, что нам действительно нужна какая-то основа мультимедиа, чтобы на ней строить наш видеоредактор. И я обратился к *GStreamer*.

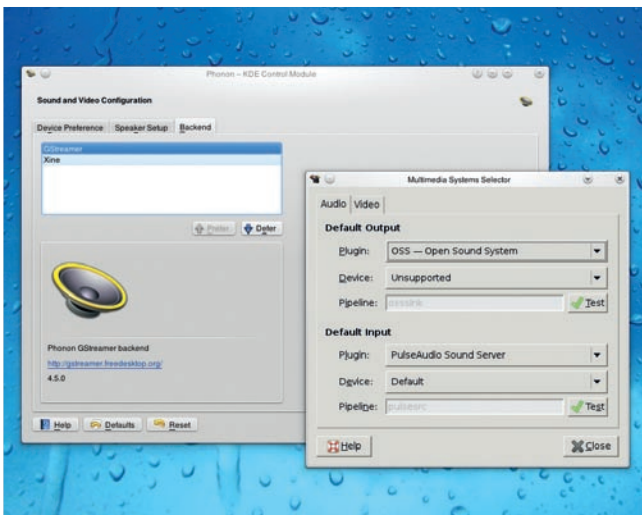
LXF: Как начинался *GStreamer*?

BT: В 1999 году Эрик Вальтинсен [Erik Walthinsen] выпустил первый *GStreamer*. Казалось, он идеально подходит для мультимедиа Trinity, так что я набросился на код и проектировал многие части ядра.

Мы начали рассылку, открыли канал IRC и создали сайт. Мы поместили код и списки рассылки на свежесозданной инфраструктуре SourceForge и получили немало заинтересованных людей – многие до сих пор активно работают в *GStreamer* или со сходными технологиями.

LXF: Каковы были ваши первоначальные цели?

BT: Цель была простой: мировое господство. А если серьезно, мы начинали примерно тогда же, что и проекты *Xine*, *MPlayer* и *VLC*. Писать код медиа-плеера было прикольно, и такого элемента явно не хватало в рабочем столе Linux. В среду мультимедиа мы хотели включить кодирование, передачу голоса по протоколу IP (VoIP) и видеоредактирование.



» Вим и его коллеги по Collabora работают над версией 0.11.

ПРО УТАН-GLX

«Джон Кармак (*Quake/Doom*) был моей первой встречей с рок-звездой программирования.»

LXF: Каковы главные особенности *GStreamer*?

BT: Мы всегда стараемся экспериментировать. Никто из нас не знает, как выглядит идеальная среда мультимедиа, поэтому мы тратим немало времени на эксперименты.

Разработка перешла на более высокий уровень, когда целая команда хакеров *GStreamer* была нанята Fluendo в 2004 году. В версию 0.10 (по-прежнему текущую) вошло множество изменений, и они выдержали испытание временем.

Я считаю, что *GStreamer* теперь можно ставить в один ряд с большинством альтернативных видеоплееров по части простоты воспроизведения, и он превосходит их расширенными функциями, такими как воспроизведение в обратном порядке и хитрыми режимами. Нам также хочется верить, что у нас один из лучших стеков RTP – это центральный компонент приложений VoIP и видеоконференций.

LXF: Какие приложения используют *GStreamer*?

BT: Взглянув на типичный состав современных рабочих столов Linux, вы увидите, что по умолчанию *GStreamer* используют все приложения.

Музыкальные проигрыватели, такие как *Rhythmbox* и *Banshee*; видеопроигрыватель *Totem*; приложение камеры *Cheese*; видеоредактор *PiTiVi*; VoIP-приложение *Empathy*; и многие другие. *GStreamer* – не отдельное

приложение, а простой в использовании API, облегчающий создание красивых интегрированных приложений.

LXF: Вы и многие другие разработчики *GStreamer* присоединились к Collabora Multimedia. Как это случилось?

BT: После релиза 0.10 мы почувствовали, что готовы начать свой бизнес консалтинга по *GStreamer*. С нами связались основатели Collabora, Роберт Маккуин [Robert McQueen] и Филипп Калаф [Philippe Kalaf], и предоставили нам финансирование и использование их инфраструктуры и имени. В мае этого года мы, наконец, объединились, и в настоящее время являемся частью Collabora Limited.

LXF: Какую роль играет Collabora в *GStreamer* сегодня?

BT: Огромную. Collabora стремится улучшить *GStreamer* и состояние открытого исходного кода. Мы также главные организаторы саммита *GStreamer* и со-организаторы Linux Kernel Summit в Праге в конце этого года.

LXF: Где вы видите будущее *GStreamer*?

BT: Сейчас мы работаем над версией 0.11, и надеемся сделать версию 1.0 в конце этого года. Мы расширяем 0.10 последние шесть лет; следующий шаг – убрать накопившийся мусор и слегка улучшить там и сяма.

Для нас также важно подогнать *GStreamer* к новым тенденциям, таким как 3D и даже видео MultiView, оказывать более широкую поддержку аппаратного ускорения в GPU или DSP на устройствах рабочего стола и встраиваемых устройствах, а также создать больше возможностей для оптимизации скорости.

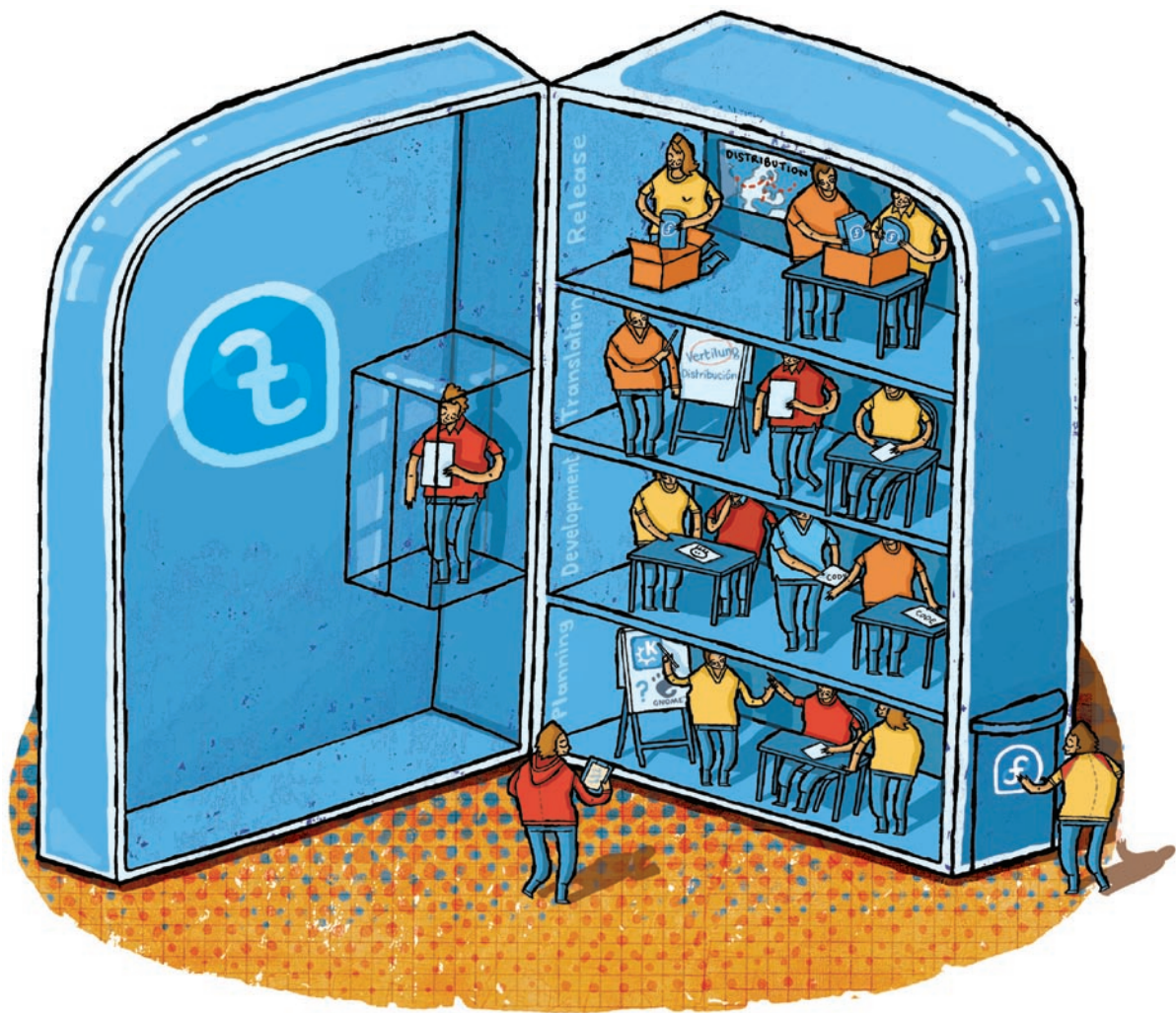
LXF: Как люди могут способствовать развитию *GStreamer*?

BT: Большинство информации можно найти на основном сайте – (gststreamer.freedesktop.org). Во-первых, прочитайте Руководство и Руководство по написанию модулей, чтобы получить представление, как все это работает, а затем останется лишь найти задачу, которую вы хотели бы реализовать/исправить. У нас, например, есть ряд задач в [bugzilla](https://bugzilla.gnome.org/).

Если вы хотите начать писать что-то или поподробнее узнать, как обстоят дела, то мы обычно сидим в IRC. Заходите на [irc.freenode.net](https://irc.freenode.net/#gststreamer) в #*gststreamer* и общайтесь с нами там. Большинство делает отчеты об ошибках и предоставляет заплатки для реализации новых функций или исправления проблем. **LXF**

Внутри Fedora

Составление дистрибутива – задача чрезвычайно сложная. Джонатан Робертс ведет нас за кулисы Fedora, чтобы мы увидели, как это происходит.



Только что вышла Fedora 15, принесла нам, наряду с другими инновационными функциями, Gnome 3, systemd и поддержку динамической настройки брандмауэра. Как бы ни впечатляюще звучал этот список, отбарабанить его легко – забыв о работе, скрытой за каждым выходом релиза.

Конечно, нам прекрасно известно, что на планирование, разработку

и тестирование каждого релиза уходят тысячи часов, но редко когда мы действительно ценим это.

Более того, мы типа в курсе, что имеются умные инструменты и процедуры, которые превращают это

в жизнь, но не особо вникаем, что это за процессы и как они работают.

Мы подумали, что люди, которые занимаются этим, а также применяемые ими утилиты и технологии более чем достойны постоять под огнями софитов.

В конце концов, это пригодится и другим создателям передовых технологий – ведь именно благодаря обмену информацией сво-

бодное программное обеспечение нас и радует.

В нашей статье мы проведем вас за кулисы Fedora и совершим прогулку от ранних стадий планирования и разработки до внесения финальных битов.

«Мы подумали, что эти люди достойны огней софитов.»

Важно отметить, что такого внимания заслуживает не только Fedora: существует огромное количество проектов свободного ПО, чудесных с инженерной и логической точки зрения. И данной статьей мы затеваем новую серию – мы ежемесячно будем заглядывать за кулисы разных дистрибутивов или проектов.

Надеемся, что по завершении этой серии вы будете включать свой компьютер с должным почтением и уважением, осознавая те усилия и умения, что заставляют его работать.

Еще до планирования или разработки конкретных деталей каждый релиз начинает свою жизнь в «rawhide [сыроматной]» ветке Fedora. Это полноценная версия дистрибутива, но она никогда не выходит официально, а, напротив, существует в непрерывной стадии перемешивания: постоянно добавляются новые пакеты, а старые, будь они важными или побочными, обновляются до свежих версий; вводятся новые функции, а устаревшие ликвидируются.

Результат предсказуем: rawhide-версия не является стабильной. Иногда ее можно установить, но и это не гарантируется; также нет уверенности, что можно загрузиться и безопасно использовать систему, составленную из пакетов этой ветки Fedora.

Вместо этого, ее цель – снабдить разработчиков полигоном, где они начинают понимать, какой эффект на остальную часть дистрибутива произведет то или иное обновление; и именно там обновления становятся доступными другим разработчикам для тестирования пакетов и программ.

Чтобы понять, как релиз Fedora собирается из частей, необходимо понять, как за шесть месяцев осуществляется его переход от хаоса до протестированной и усовершенствованной коллекции пакетов, которую можно считать стабильной. Нашим отправным пунктом станет понимание того, как создается rawhide.

Коллекция пакетов

Во-первых, вам нужно знать, что Fedora – это дистрибутив, основанный на RPM, то есть по сути это коллекция «пакетов» – скомпилированных двоичных файлов, укомплектованных с дополнительными «метаданными» (данными о данных), чтобы помочь компьютеру автоматически отслеживать, какие файлы содержатся в пакете и каким программам они принадлежат. Благодаря этому установку, обновление и удаление пакетов через менеджер пакетов производить очень легко.

Вообще говоря, RPM бывают двух типов: RPM исходников (SRPM) и стандартные RPM. Первые отличаются тем, что содержат не двоичные файлы, а исходные файлы программ, а также дополнительные заплатки и набор инструкций для компиляции исходного кода в стандартный RPM.

Именно так все пакеты Fedora, и все функции Fedora, начинают свою жизнь: разработчики пишут «файл спецификации» и прилагают его к исходному коду, а затем запускают программу, создающую легко устанавливаемый RPM. После того, как пакет написания и скомпилирован, первой задачей Fedora Project становится внести его в управляемую систему для отслеживания изменений. Для этого нужен git.

Каждый пакет в Fedora – по крайней мере, RPM исходников (SRPM) – хранится в собственном git-репозитории. Если кто-нибудь добавляет новый пакет или обновляет старый, он должен создать новый репозиторий или «передать» обновление старому.

Для тех, кто не знает, сообщим, что git – это распределенная система контроля версий [distributed revision control system]. Кратко, это небольшая программа, которая отслеживает различные версии («ветки») других программ.

В Fedora git-репозиторий каждого пакета обычно содержит одну ветку для каждой доступной на данный момент версии дистри-

бутива. Так как rawhide – тоже версия Fedora, самую последнюю версию каждого пакета всегда можно найти в ветке rawhide его git-репозитория.

Итак, rawhide начинает свою жизнь как набор git-репозитория, каждый из которых содержит ветку «rawhide» с самыми последними SRPM. Не каждый может отправлять файлы в эти репозитории – чтобы получить разрешение, нужно иметь рекомендацию и успешно пройти процедуру рассмотрения; однако тем, кто заработал это право, предоставляется значительная свобода, о чем говорит тот факт, что rawhide постоянно меняется и обновляется.

Однако не всем это подходит, поскольку установка SRPM требует немалого труда, и это явно не та методика, ко-

торую хочется применить для установки полностью рабочей системы. Более того, у людей здесь нет никакой возможности узнать про самые последние изменения в отдельных SRPM git-репозиториях, как это делается в менеджере пакетов дистрибутива.

Чтобы привести rawhide в более удобное состояние, Fedora Project полагается на две специальные утилиты.

Koji и Mash

Первая из них – «система сборки», известная в Fedora как Koji. Она используется для создания из мешанины исходного кода, спецификаций пакета и заплат, хранящейся в git, легко устанавливаемых – но не обновляемых автоматически – файлов RPM.

Koji всегда вызывается с определенной веткой git, которая сообщает, для какой версии Fedora этот пакет предназначен. Это необходимо, поскольку если пакет собран в присутствии определенной версии других важных пакетов, то он должен запускаться тоже в их присутствии.

Koji, поступает по уму и создает окружение chroot – виртуальную файловую систему, для гарантии, что пакет собран с версиями важных пакетов, доступными в той версии Fedora, для которой пакет готовится.

Итак, Fedora Project запускает Koji со всеми файлами в «rawhide»-ветках своих проектов, создавая набор легко устанавливаемых RPM. Мы еще не пришли к rawhide,



» В логотипе Fedora Project «четыре F» (Свобода, Дружба, Функции, Первенство) отражены ее ценности.

Открыта нараспашку

В нашей статье мы обсудим по крайней мере полдюжины различных утилит, заменяемых командой Fedora для компиляции релизов. Все они – открытое ПО, и таковыми же являются подробные документы, описывающие, как Fedora вводит их в дело.

Это значит, что каждый может скачать и изучить их, и узнать, как с помощью утилит и процедур промышленного масштаба создать собственную коллекцию пакетов, дополняющую Fedora, или даже собственный дистрибутив, основанный на RPM. Утилиты Fedora делают эту возможность далеко не теоретической: репозитории RPM Fusion и Extra Packages for Enterprise Linux (EPEL) применяют все эти средства,

предоставляя пользователям те пакеты, что Fedora и Red Hat Enterprise Linux не могут или не хотят предоставлять.

Такая открытость, не только в пакетах, но и в утилитах и процедурах, является одним из основополагающих качеств Fedora и привлекла к проекту множество людей. Дэн Гилмор [Dan Gilmore], нынешний лидер команды Release Engineering, прежде чем стать куратором системы сборки, помогал команде Fedora Infrastructure, а потом ушел в Red Hat. Несмотря на то, что это теперь его штатная должность, в результате всесторонней и открытой природы средств, предоставляемых Fedora, он, по его словам, все еще чувствует «радость и гордость от своего участия».

FUDCon

Хотя многим не нравится тесная связь Fedora со своим коммерческим спонсором Red Hat, но она определенно имеет свои преимущества. Одно из них – у компании есть средства на проведение регулярных, по-настоящему международных конференций. Их называют Fedora User and Developer Conferences (FUDCon), и это прекрасный способ скоординировать новые возможности, накопать множество кода и выпить много пива.

Что здорово в FUDCon – это то, что они проводятся по всему свету: регулярные события происходят в Латинской Америке, Северной Америке и в Европе. В этом году также были выделены деньги на проведение FUDCon в Азиатско-Тихоокеанском регионе, так что явление становится поистине глобальным.



Участники FUDCon на групповом снимке.

Фото: Никку Букунев (Nicu Buculei)

так как у нас пока не предусмотрен простой способ отслеживать и извлекать изменения.

Этой последней задачей занимается утилита под названием *Mash*, которая просто сгребает все файлы, накопившиеся в *Koji*, и помещает их в некое каноническое место, известное как репозиторий. В репозиторий добавляется некоторое количество метаданных, сообщающих, какая версия у размещенных пакетов самая последняя, чтобы менеджеры пакетов могли работать быстро и эффективно.

Репозиторий также содержит файл **boot.iso**; с его помощью можно запустить виртуальный компьютер, скачать требуемые файлы и установить их на реальном компьютере.

И – вуаля! Rawhide готов! Чтобы добраться до этого места, уже проделано много работы, однако до конечного продукта – стабильной, надежной операционной системы для каждодневного использования – еще далеко. Все это происходит за время шестимесячного цикла релизов Fedora Project, и начинается с двухэтапного процесса планирования.

Сначала команда Release Engineering устанавливает расписание для преобразования rawhide в очеред-

ной стабильный релиз, содержащее несколько ключевых пунктов, включая заморозку функций [feature freeze] и ветки [branch freeze], а также альфа- и бета-релизы.

Каждый из этих ключевых пунктов вносит новые ограничения на приемлемость обновлений, и по мере приближения даты релиза становится более консервативным. Именно это постепенное изменение позволяет преобразовать изменчивый rawhide в надежную стабильную систему.

Блистательные функции

В другой аспект процесса планирования вовлечен комитет Fedora Engineering Steering Committee (FESCo), определяющий, какие «функции» войдут в каждый релиз. Согласно Робену Бергеро [Robyn Bergero], программному руководителю Fedora, все функции разделяются на две обширные категории: «новые и яркие» и те, от которых зависит «критический путь» – основные компоненты, влияющие на множество других пакетов и пользователей.

Первые дают пользователям представление о новых функциях, которые появятся после обновления. Вторые зачастую включают новые пакеты, устанавливаемые по умолчанию или заменяющие старые пакеты; изменения такого типа нелегко выполнить правильно, а если они пойдут не в ту сторону, то эффект от этого затронет многих. Отслеживая их как функции, три главные инженерные команды Fedora (FESCo, Release Engineering и Quality Assurance) могут пристально наблюдать за прогрессом, проверяя, что их амбиции реалистичны, что все довольны и что на случай провала имеются резервные планы.

Несмотря на столь всестороннее планирование, Fedora остается очень меритократичным проектом. Новую функцию может предложить каж-

дый, и каждый может внести пакет в дистрибутив. Последнее слово остается за FESCo, но его состав на 100 % выбирается сообществом. Вот что сказал о политике введения новых обновлений в Fedora нынешний председатель FESCo Кевин Фенци [Kevin Fenzi]: «Мы должны прислушиваться ко всему и преобразовывать в то, что все мы считаем верным».

После принятия решения, какие функции войдут в релиз, разработчикам даются первые два месяца цикла на работу над rawhide. После этого проект входит в стадию заморозки «ветки» и «заявленных функций», когда для каждого отдельного пакета создается новая ветка, и контроль над репозиториями переходит от *Mash* к другой программе, под названием *Bodhi*.

Bodhi следит за тем, что делает *Koji*, и вместо того, чтобы решать новым или обновленным пакетам отправляться прямо в репозиторий новой ветки, помещает их в тестируемый репозиторий. Чтобы пакет мог перейти оттуда в основную, стабильную

«Несмотря на планирование, Fedora – меритократичный проект.»



Герои инфраструктуры Fedora

Вы, наверное, заметили, что многие из утилит Fedora – web-службы, предоставляющие централизованное хранение и мощности, чтобы обеспечить выполнение работ по проекту, распределенному глобально между многими разработчиками.

Без этих серверов и утилит, размещенных на этих серверах, не было бы и проекта Fedora. То есть обеспечение

стабильности и безопасности – не просто сложная задача, но и жизненно важная.

Несмотря на сложность, вся инфраструктура поддерживается в открытой и меритократической манере, как и код, и утилиты проекта. Хотя проект не разрешает всем и каждому скачивать по серверу с полными правами доступа, он поощряет изучение прин-

ципов работы системы и предложения по улучшению его работы.

Когда пользователь пройдет это и станет надежным, заслуживающим доверия и компетентным, проект в конце концов разрешит ему стать администратором своей системы.

Этот способ доказал свою эффективность, и администраторы – как добровольцы, так и от Red Hat – рас-

сеяны по всему свету. Раньше все было неформально – люди просто встречали в почтовые рассылки и спрашивали, как они могут помочь; сейчас, однако, проект формализован по схеме Infrastructure Apprentice. Это прекрасный способ изучить работу систем в реальном мире для студентов, и один из способов стать участником.

коллекцию пакетов, он должен либо получить положительный отзыв, либо продержаться определенное время без отрицательных отзывов.

Это важно для перехода Fedora от хаотической смеси к стабильности, пригодной для каждодневного использования.

Альфа, бета... выйди вон

После этого проект выпускает два пререлиза (альфа и бета); оба содержат все запланированные функции и распространяются в виде установочных DVD или live CD. Для их создания используются более специальные утилиты, соответственно *Pungi* и *Live CD Creator*, и перед выходом релиза созывается собрание трех основных команд разработчиков, убеждающееся, что соблюден минимальный набор стандартов.

И снова, идея таких пререлизов – затребовать дальнейшую обратную связь от максимального числа пользователей, чтобы финальный продукт был стабилен и полон самых свежих и лучших усилий разработчиков за последние шесть месяцев. Они особенно удобны, так как предоставляют общую платформу, на которой команда QA строит свои тестовые дни. Это значительные события в сообществе: разработчики функций и опытные инженеры заходят в Internet Relay Chat (IRC), чтобы провести пользователей через серию испытаний на их системах. В каждый день тестирования внимание уделяется функции или другому критическому компоненту, что позволяет получить большое число откликов.

Это дает Fedora возможность протестировать и отработать новые функции на максимально широком спектре оборудования, и представляет пользователям, менее продвинутым технически, отличные шансы посылить помощь разработке. Все, что для этого требуется – копия самого последнего пререлиза и доступ к IRC, а разработчики протаскают вас через остальное.

Интересно, но якобы самое большое испытание для этих релизов – распространение гигабайтов данных по всему свету – является простейшей частью операции.

Команда Release Engineering собирает все пакеты из той версии репозитория, что была успешно скомпилирована на носитель с установкой, а затем просто задает права на коллекцию такими, чтобы она не была доступна из внешнего мира.

Затем они разрешают доступ партнерам – владельцам зеркал, каждый из которых управляет клоном главного репозитория. В результате пользователи зря не осаждают центральный сервер: им доступны сотни серверов по всему миру, откуда можно скачать релиз.

Колдовства здесь нет: все зеркала просто используют *rsync* для создания своих копий главного зеркала Fedora. Когда все это



сделано и настает день релиза, на всех зеркалах просто меняют права доступа («опрокидывают биты»), чтобы открыть его всем – и процесс скачивания пошел!

➤ Внутренняя арт-группа создала собственный конверт для DVD.

«Дни тестов – шанс помочь для непродвинутых пользователей.»

В итоге Fedora Project ухитряется из хаотической смеси пакетов, содержащих самый свежий код, изготовить тщательно протестированную, крепко сбитую коллекцию и в несколько приемов доставить ее по всему свету. Остается только составить носитель для окончательного релиза, который называется релизом GA (General Availability – общедоступным), и поставить его на зеркала тем же способом, что применялся для альфа- и бета-релизов.

Хотя, конечно, остается и еще одно дело – насладиться заслуженным банкетом, когда пользователи и разработчики всего мира празднуют выход релиза. **LXF**



Больше, чем дистрибутив

В данной статье основное внимание было уделено техническим процедурам, используемым при создании релиза дистрибутива Fedora.

Однако Fedora – это не просто код, и в каждый релиз вовлекается работа многих других команд.

Сюда входит арт-группа, а также команды документации, маркетинга, переводов и сайта; каждая из них дает тот или иной вклад в релиз.

Арт-группа, например, создает обои рабочего стола, используемые по умолчанию, и сопровождающие материалы для брендинга процессов загрузки и установки, а команда документации гарантирует, что информация о релизе всесторонняя.

Все эти команды пользуются во многом теми же утилитами, что и программисты, включая системы контроля версий, и вся работа проходит в открытом формате, чтобы каждый смог сделать свой вклад.



➤ Обои по умолчанию в рабочем столе Gnome 3 в Fedora 15.



Как нам «легализовать GPL»?

Павел Протасов рассматривает возможные изменения российского законодательства, имеющие отношение к свободным лицензиям.

Тема соответствия российскому законодательству типовых лицензий на программы с открытыми исходными текстами и другие произведения является «больной» для российского OpenSource-сообщества: слишком много вокруг нее разведено спекуляций. Применением GPL заинтересовались не только разработчики и пользователи свободного ПО, но при этом в умах населения все еще продолжает циркулировать слух, что лицензия «не является законной, поскольку государство ее не одобряет».

Политика открытого лицензирования

На беспочвенности этого слуха мы уже останавливались (в статье «Восемь “нет” о GNU GPL», LXF139-140). Гражданский кодекс закрепляет принцип «свободы договора», разрешая заключать сделки, даже не предусмотренные законодательством, так что никакого «одобрения» и не требуется.

Однако миф о том, что «GPL в России недействительна, поскольку не одобрена государством», оказался на удивление живучим. Пользуясь им, продавцы коммерческого ПО пугают потенциальных клиентов тем, что «придет милиция и все заберет». (Что самое смешное – слухи о злой милиции сильно преувеличены, и при проверках «линухсы» ее, как правило, не интересуют).

Еще одна группа паразитов на этом мифе – так называемые «эксперты по SAM». «SAM», или «Software Asset Management» – это «управление лицензиями на ПО». Зародилась эта дисциплина из предельно запутанных «правил лицензирования» от Microsoft. «Эксперты по SAM» всегда будут рады предостеречь своих клиентов от необдуманных поступков типа срывания «лицензионной наклейки на Windows» с корпуса компьютера – ведь это по правилам Microsoft делает установленную копию ОС «нелегальной», и ее придется покупать снова. (К тем же самым последствиям, по словам технической поддержки Microsoft, приводит и выбрасывание коробки от дисков с ПО, поскольку у «коробочных» версий «лицензионная наклейка» прилеплена именно на нее).

Учитывая то, что многие «эксперты» по совместительству программами еще и торгуют, предельно простые условия GPL и других свободных лицензий для них – что нож вострый. Единственный выход – пусть лучше такие лицензии будут «нелегальными». Потому что «не одобрены государством»...

Но в последнее время политические события вокруг свободных лицензий активизировались, причем на самом высшем уровне. Еще в апреле Президент провел встречу с представителями интернет-сообщества в Российской государственной библиотеке для молодежи. В июне по итогам этой встречи органам власти было дано поручение подготовить проект изменений в законодательстве, направленных на расширение возможностей по свободному использованию произведений.

Кроме того, в конце июня Медведев лично поучаствовал в загрузке на сервер «Викисклада», хранилища свободно распространяемых произведений, фотографий, которые РИА «Новости» решило подарить сообществу на условиях лицензии Creative Commons Attribution-ShareAlike 3.0 (CC-BY-SA).

После этого всем, по идее, должно быть ясно, что курс на либерализацию копирайта взят устойчивый...

Поручение Президента

«Подготовить предложения о внесении изменений в гражданское законодательство Российской Федерации, направленных на закрепление для авторов произведений возможности предоставлять свои произведе-

ния на условиях свободных лицензий неограниченному кругу лиц (аналогичных Creative Commons, GNU FDL).

Ответственные: Щёголев И. О., Яковлев В. Ф. Срок – 1 августа 2011 г.»

Проект изменений в статью 1233 ГК

6. Правообладатель может сделать заявление о предоставлении любым лицам возможности безвозмездно использовать принадлежащий ему результат интеллектуальной деятельности на определенных им условиях и в течение указанного им срока. В течение этого срока любое лицо вправе использовать этот результат на указанных условиях.

При отсутствии в заявлении правообладателя указания на срок считается, что этот срок составляет пять лет.

При отсутствии в заявлении правообладателя указания на территорию считается, что это территория Российской Федерации.

Правообладатель не имеет права осуществлять такие действия при наличии действующих возмездных лицензионных договоров, по которым предоставляется право использовать результат интеллектуальной деятельности в тех же пределах.

Указанное заявление должно быть сделано путем размещения на официальном сайте федерального органа исполнительной власти по интеллектуальной собственности.

В случае, если в отношении результата интеллектуальной деятельности осуществлена государственная регистрация, сведения о таком заявлении должны

быть внесены федеральным органом исполнительной власти по интеллектуальной собственности в соответствующий государственный реестр.

В течение срока действия заявления не может быть отозвано, а предусмотренные в нем условия использования не могут быть изменены.

Положения настоящего пункта не применяются к результатам интеллектуальной деятельности, в отношении которых настоящим Кодексом установлен порядок предоставления правообладателем открытой лицензии (статьи 1368 и 1429).

Но уже сейчас существует другой широко обсуждаемый законопроект, который содержит положения, направленные на «легализацию» свободных лицензий. Проект вносит изменения не только в «копирайтные» статьи четвертой части ГК, но и в другие разделы, причем изменения весьма существенные. Поэтому за его обсуждением следят юристы, которые специализируются в самых разных отраслях права. Давайте займемся этим и мы.

Как бы свободные...

Проектом предусмотрено добавление в статью 1233, описывающую распоряжение исключительным правом на результаты интеллектуальной деятельности,

шестого пункта, который и «легализует свободные лицензии». Сама статья 1233 содержится в той главе четвертой части, которая определяет общие положения, применимые

ко всем результатам интеллектуальной деятельности, а не только к произведениям, охраняемым авторским правом.

Добавленный пункт предусматривает «заявление» правообладателя о предоставлении любому лицу возможности безвозмездно использовать свою «интеллектуальную собственность» на определенных условиях и в течение определенного срока. Если срок не указан, он считается равным пяти годам. Заявление является безотзывным, а условия использования нельзя изменить. Такие заявления размещаются на сайте «федерального органа исполнительной власти по интеллектуальной собственности», то есть Роспатента.

Последнее требование повергло в ужас всех наблюдателей от IT-сообщества. По понятным причинам: Роспатент – он только и исключительно российский, а произведения по свободным лицензиям распространяются во всем мире.

Но беда такого порядка «освобождения» произведения – не только в этом. Попытка ввести в закон некое «заявление» не согласуется с некоторыми другими статьями ГК, а также с одним из основных принципов авторского права.

Принцип этот закреплен в четвертом пункте статьи 1259 ГК, и звучит он так: «Для возникновения, осуществления и защиты авторских прав не требуется регистрация произведения или соблюдение каких-либо иных формальностей».

Именно так и начиналось авторское право: «Статут королевы Анны», который считается первым законом о копирайте, предусматривал обязательную регистрацию произведений. Права на него действовали в течение четырнадцати лет, после чего их можно было продлить еще на четырнадцать. Таким образом,

осуществлять их автор мог только после выполнения некоторых формальностей.

Со временем список формальностей становился все меньше; последней стал всем хорошо известный «знак копирайта», латинская буква «С» в окружности. Сейчас такой знак проставляется по желанию, однако сравнительно недавно его использование было обязательным. Оно требуется Всемирной конвенцией об авторском праве, но в странах, присоединившихся к Бернской конвенции, такого требования уже нет.

Одним из основных принципов Бернской конвенции является «принцип автоматической охраны», который как раз и состо-

ит в том, что авторские права возникают автоматически, а автор может использовать свое произведение без соблюдения каких-либо формальностей.

Этому основному принципу и противоречат планируемые

поправки, которые связывают осуществление прав на свободно распространяемые произведения с заявлением, которое, к тому же, и опубликовано должно быть на конкретном сайте. То есть – с формальностью...

Еще непонятно, почему по такой схеме распространять свои произведения можно только безвозмездно. Авторы поправок при их написании, похоже, держали в голове только лицензии CC и FSF. А если автор хочет денег? Он, получается, будет лишен возможности предоставить свое произведение «любимым лицам»? И почему такая дискриминация? Да, непонятно...

Ну и, наконец, еще одно противоречие. ГК в статье 1269 предусматривает так называемое «право на отзыв»: автор произведе-

«Гражданский кодекс закрепляет принцип “свободы договора”»

Две основных конвенции

В сфере авторского права действуют две главных международных конвенции. Первая из них, «Бернская», предоставляет более высокий уровень охраны произведений, хотя и была принята раньше. Вторая, «Всемирная», или, как ее иногда называют, «Женевская», принималась для того, чтобы к ней могло присоединиться как можно большее число государств, включая те, в которых законом предусмотрены ограничения, не совместимые с Бернской конвенцией.

Именно по этой причине Всемирная конвенция «подстраивается» под внутреннее законодательство стран, допуская многочисленные послабления в режиме охраны. Проставление знака охраны авторских прав – всего лишь одно из таких послаблений. В то же время, согласно Бернской, более жесткой конвенции, наоборот, требуется, чтобы внутреннее законодательство присоединившейся страны было приведено в должное соответствие с текстом конвенции.

Из Всемирной конвенции об авторском праве

Статья III, 1. Любое Договаривающееся Государство, которое в соответствии со своим внутренним законодательством требует в качестве условия охраны авторского права соблюдение таких формальностей, как депонирование экземпляров, регистрация, оговорка об оставлении за собой прав, нотариальные удостоверения, уплата сборов, про-

изводство или выпуск в свет на территории данного государства, будет считать эти условия выполненными в отношении всех произведений, пользующихся охраной по настоящей Конвенции и впервые выпущенных вне территории этого Государства и автор которого не является одним из его граждан, если, начиная с первого выпуска этого произведения,

все его экземпляры, выпущенные с разрешения автора или другого лица, обладающего авторским правом, будут иметь знак © с именем лица, обладающего авторским правом, и с указанием года его первого выпуска; знак, имя и год выпуска должны быть указаны таким способом и на таком месте, чтобы было ясно видно, что права автора охраняются.

дения, который уже его обнародовал, может от такого обнародования отказаться и даже изъять из обращения уже выпущенные экземпляры произведения. При этом он должен возместить причиненные таким решением убытки. В случае, если автор опубликовал свою работу в Интернете, такой отзыв реализовать очень легко: она просто удаляется с сайта. Правда, «закачать обратно» уже скачанное не получится...

Правом этим пользуются сравнительно редко, а к программам оно не применяется вообще, но совершенно непонятно, почему авторы поправок лишают возможности «отзыва» тех авторов, которые изъявили желание воспользоваться «свободной лицензией а-ля ГК».

При изучении «копирайтного» законодательства нетрудно заметить, что проблемы у свободных лицензий возникают из-за концептуальной их несовместимости с ГК. Кодекс исходит из того, что создание произведения и передача прав на него – это некий серьезный шаг. За нарушение авторских прав предусмотрена совершенно драконовская «компенсация» с минимальным размером в десять тысяч рублей. Дополнительные меры по защите автора предусмотрены и при заключении договора, который должен быть только письменным («упрощенный порядок» заключения такого договора для ПО – это тоже разновидность письменной формы договора).

Если срок действия договора не указан, то он принимается равным пяти годам. Если не указана страна, в которой действует договор – он действует только на территории России. Эти общие правила, как вы видите из текста проекта, в нем даже продублированы дополнительно.

Такие меры призваны защитить права авторов, но, как показывает практика, своего назначения они не достигают: если автора нужно обмануть, его обманут все равно. И даже обманывать не требуется: посредники в лице «лейблов» и издательств в договоре являются более сильной стороной и способны навязать автору свои условия...

В то же время, существуют многие тысячи творцов, которые поют, играют, рисуют и программируют, не думая о прибыли, а творчество свое хотят раздавать всем желающим бескорыстно или в качестве «рекламных материалов». Но «законно» этого сделать они не могут, причем основным препятствием для этого являются как раз те положения закона, которые призваны их защищать...

Да и вообще, регулирование копирайтной сферы у нас, похоже, движется из крайности в крайность. Сначала законодатель принимает четвертую часть ГК, в которой за-прещает все, что можно, а затем – пытается ввести «свободные лицензии», но теперь отбирает часть прав уже у автора...

Между тем, гражданское законодательство уже предусматривает «открытые лицензии» на изобретения и селекционные достижения, при помощи которых правообладатель может передать

право на их использование любому желающему. Они описаны в статьях 1368 и 1429. Статьи эти упомянуты и в тексте законопроекта, из которого видно, что авторы конструировали новую норму по аналогии со статьями об «открытых лицензиях», и из них же взяли требование о регистрации произведения в Роспатенте: и для изобретения, и для селекционного достижения это – обязательные условия.

Существование «открытых лицензий» в отечественной правовой системе – еще один из аргументов в пользу того, что GPL и ей подобные типовые договоры могут применяться уже сейчас. Ведь ГК в своей шестой статье разрешает применение законодательства «по аналогии», то есть, применение сходных норм права в том случае, если прямо какие-то правоотношения не урегулированы.

Ну, а если нужно более подробно описать применение свободных лицензий в законе – наверно, стоит заменить «публикацию на сайте Роспатента» на просто «публикацию», установив в качестве ее условия возможность достоверно определить, от кого такая публикация исходит. Перенести статью из «общей» главы в следующую, семидесятую, которая содержит нормы, распространяющиеся исключительно на авторское право. Убрать бесплатность и безотзывность, разумеется. И не нужно называть лицензионный договор «заявлением», плода сущности без необходимости...

Русский DMCA

Еще одно изменение, которое можно принять за «либерализацию», планируется внести в статью 1299 ГК. В ней описаны так называемые «технические средства защиты авторских прав», то есть различные «защиты от копирования», средства контроля доступа к произведению и тому подобное. Сейчас, согласно этой статье, в случае удаления пользователем такой защиты правообладатель получает право требовать от пользователя возмещения убытков или уплаты той самой драконовской компенсации с минимальным размером в десять тысяч.

Законопроектом планируется добавить пункт, в соответствии с которым в некоторых случаях, когда защита мешает «свободному использованию» произведения, пользователь вправе требовать от правообладателя либо удалить ее, либо обеспечить

возможность такого использования как-то еще.

Но тот, кто думает, что разработчики проекта нас с вами благодетельствовали, просто не знает, как выглядела первая редакция этой ста-

тьи. А ведь в ней пользователю давалось право самому «сломать» защиту, которая мешала свободному использованию, а не ходить на поклон к правообладателю (см. врезку).

Как мы видим, в самом первом варианте статьи возмещение убытков или выплата компенсации исключалась в любом случае, когда ГК разрешал использование произведения без согласия ав-

«GPL и ей подобные договоры могут применяться уже сейчас.»

Изменения в статье 1299 ГК

2. В отношении произведений не допускается:

- 1) осуществление без разрешения автора или иного правообладателя действий, направленных на то, чтобы устранить ограничения использования произведения, установленные путем применения технических средств защиты авторских прав;
- 2) изготовление, распространение, сдача в прокат, предоставление во временное безвозмездное пользование, импорт, реклама любой технологии, любого технического устройства или их компонентов, использование таких технических средств в целях получения прибыли либо оказания соответствующих услуг, если

в результате таких действий становится невозможным использование технических средств защиты авторских прав либо эти технические средства не смогут обеспечить надлежащую защиту указанных прав.

3. В случае нарушения положений, предусмотренных пунктом 2 настоящей статьи, автор или иной правообладатель вправе требовать по своему выбору от нарушителя возмещения убытков или выплаты компенсации в соответствии со статьей 1301 настоящего Кодекса, ~~кроме случаев, когда настоящим Кодексом разрешено использование произведения без согласия автора или иного правообладателя.~~

4. *В случае, когда статьями 1274 и 1278 настоящего Кодекса разрешено использование произведения без согласия автора или иного правообладателя и такое использование невозможно осуществить в силу наличия технических средств защиты авторских прав, лицо, желающее осуществить такое использование, может требовать от автора или иного правообладателя снять ограничения использования произведения, установленные техническими средствами защиты, либо предоставить возможность такого использования по выбору правообладателя.*

[Вычеркнута удаленная часть, курсивом выделено то, что планируется добавить.]

тора. Статей с такими разрешениями в Кодексе целых восемь. Однако в октябре прошлого года статья 1299 была изменена и это исключение отсюда убрали.

Статьи 1274 и 1278 описывают использование в информационных, учебных, научных и культурных целях, а также в целях правоприменения. В случае, если происходит создание копии в личных целях (ст. 1273) или копирование в библиотеке (ст. 1275), права «требовать» чего-то у правообладателя пользователь иметь не будет. И, самое главное, проектом так и оставлено запрещенным удаление защиты с компьютерных программ, свободное использование которых предусмотрено в статье 1280 ГК.

Ну, и в довершение всего достаточно представить, как пользователь будет «требовать» чего-то у правообладателя, чтобы понять, что планируемое нововведение – это даже не подкачка, а просто декларация, выполнить которую на практике будет невозможно.

Правда, кое-какие полезные для свободных лицензий

нововведения в проекте все же содержатся. К примеру, вносится ясность относительно того, запрещены ли безвозмездные лицензионные договоры в отношениях между коммерческими организациями.

Еще в конце 2008 года в одном из веблогов были опубликованы ответы на депутатские запросы из Министерства финансов и Исследовательского центра частного права, которые касались использования бесплатного ПО в организациях. И Минфин, и Исследовательский центр в своих разъяснениях написали, что безвозмездная лицензия – не что иное, как дарение, которое между коммерческими организациями запрещено. И посеяли этим в умах еще один миф, не менее вредный, чем миф о «незаконности GPL».

Задача такого запрета на дарение – пресечь возможные злоупотребления, когда под его видом из компании выводятся активы, например, при банкротстве. Но такая ситуация в корне отличается от безвозмездного предоставления всем желающим права на использование программы. Поэтому распространившиеся после публикации писем опасения того, что налоговики могут придаться и оценить свободное ПО «по стоимости коммерческих аналогов» не имели под собой абсолютно никаких оснований. (Это, кстати, невыгодно самим налоговикам: стоимость ПО в таком случае подлежала бы исключению из прибыли, что повлекло бы за собой снижение суммы налогов).

Авторы проекта вводят в статью 1235, посвященную лицензионному договору, новый пункт, в соответствии с которым коммерческим организациям запрещается только безвозмездное

предоставление друг другу исключительных лицензий, то есть таких, которые могут предоставляться только одному лицу и давать ему полный контроль над использованием произведения. Лицензии неисключительные можно «дарить» друг другу сколько угодно. Именно на условиях неисключительной лицензии программисты передают пользователям бесплатное ПО, поэтому можно будет не опасаться «придинок налоговой» после принятия проекта.

Еще существенно изменяется статья 1286, в той части, которая описывает «упрощенный порядок» заключения лицензионного договора на использование программного обеспечения. Порядок этот заключается в том, что условия договора излагаются на экземпляре ПО или его упаковке. Начало использования программы

считается заключением договора. В новой редакции статьи такой договор «по умолчанию» считается безвозмездным и бессрочным, свое действие он прекращает с момента отчуждения экземпляра его владельцем.

Казалось бы, в самый раз для свободного ПО, но... «экземпляр» – это материальный носитель с программой. Норма эта применима только к программному обеспечению, распространяемому на дисках...

В целом – чтение проекта поправок с мысленным налогом на существующую практику распространения свободного ПО и произведений по СС-лицензиям оставляет двойственные впечатления. С одной стороны, авторы хотели как лучше. С другой – почему-то упорно вспоминается дорога вы поняли куда, которую мостят вы поняли чем... **LXF**

«Кое-какие полезные нововведения в проекте все же содержатся.»

ССЫЛКИ

- » Репортаж о встрече с представителями интернет-сообщества: <http://www.kremlin.ru/news/11427>
- » Текст поручения Президента: <http://www.kremlin.ru/news/11427>
- » Проект изменений в ГК на сайте Минэкономразвития: http://www.economy.gov.ru/minec/activity/sections/corpmanagement/civil_code/full_text_civil_code/140411_gk#
- » Статья о «Статуте королевы Анны»: http://ru.wikipedia.org/wiki/Статут_королевы_Анны
- » Письма Минфина и Исследовательского центра частного права: <http://bablaw.livejournal.com/45358.html>

20 способов сломать Linux

Linux славится надежностью, но **Нейл Ботвик** доказывает, что все равно есть море способов повредить идеально работающую систему.

01 Переполнение файловой системы

Если файловая система, содержащая `/var`, заполнится, ни одной программе будет не записать сообщение в журнал, и все системные процессы заглохнут. Замусорить `var/log` способен пошедший вразнос процесс; а если вы все храните в одной файловой системе, вспомните, сколько вы наскладывали в свой домашний каталог.

02 Переустановка Windows

Нет, это не типовой анти-Microsoft'овский выпад линуксоида – установщик Windows не особо дружелюбен к иным операционным системам и загрузчикам, и если вы переустановите Windows, он ваш загрузчик затопчет. Впрочем, нужды в переустановке Linux нет: установочный CD обычно предоставляет опцию исправления загрузчика, или можно запустить `grub-install` с Live CD.

03 Нехватка памяти

4 ГБ ОЗУ – вроде и много, но только пока вы не отдали какую-нибудь безумную команду: например, `gimp *.jpg` в каталоге, набитом фотографиями с разрешением 12 Мп (не спрашивайте, откуда я это знаю). Вас спасет раздел подкачки, но все будет так тормозить, что создастся впечатление системного сбоя. По исчерпанию раздела подкачки сбой, кстати, тоже реален.

«Опасный совет может быть дан из самых добрых побуждений.»

04 Следование советам из Интернета

Web-форумы полны ценных советов и рекомендаций, но среди них встречаются потенциально опасные, способные нанести серьезный урон. Различать тут трудно: даже опасный совет может быть дан из самых добрых побуждений, а хороший совет – быть неверно сформулированным. Особой осторожности требуют команды, применяющие `su` или `sudo`.

05 Установка еще одного дистрибутива Linux

Это аналогично ранее описанной переустановке Windows. Хотя установщики дистрибутивов умеют обнаруживать Windows и создают систему с двойной загрузкой, некоторые из них не замечают на диске Linux-собратьев, и ранее установленные дистрибутивы могут исчезнуть из меню. Но сам-то дистрибутив никуда с диска не денется.

06 Нечастые обновления

Если вы месяцами не притрагивались к своему дистрибутиву, а потом затеяли установить более 300 обновлений сразу, это не обязательно вызовет поломку. Но уж если вызовет – удачи вам в поиске причины. Обновления часто и помалу – вот безопасный способ поддержания системы в актуальном состоянии.

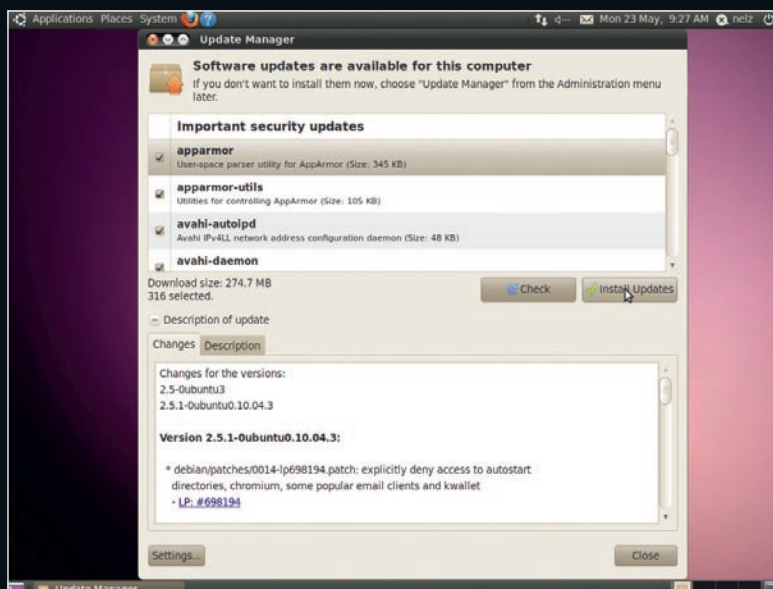
07 Обновление вслепую

Никогда не выбирайте обновление списком: тогда при сбое вы вряд ли поймете, где проблема. Или выведете список установленного ПО с помощью менеджера пакетов. Так вы узнаете, что менялось, когда сбой возник – или просмотрите файл журнала менеджера пакетов.

08 Работа с новейшим ПО

Да, все мы любим пробовать новинки, но помните о сопутствующем риске. Новое ПО не успело получить долгосрочного тестирования, и фактически теперь тестировщиком становитесь вы – поэтому не обновляйтесь до новейших альфа-релизов, если вам дорога бесперебойность работы компьютера.

Одновременная установка более 300 пакетов не обязательно сломает вам систему – но если сломает, причину сбоя найти будет нелегко.



09 Стирание /home

Это не повредит Linux, но вы потеряете все свои настройки и персональные файлы, которые там хранились. Поэтому практичнее всего держать /home на отдельном разделе, чтобы установки и обновления не портили личные настройки и данные.

10 Потеря пароля

Забыв пароль, вы не сможете ничего делать, если только в системе не установлена автоматическая регистрация. При потере пароля root компьютером вы пользоваться сможете, но вам не удастся устанавливать и обновлять программы или модифицировать систему. Потеря пароля к зашифрованной файловой системе означает возврат к расчетам в столбик.

11 Установка нового ядра

На первый взгляд это кажется странным: ведь новое ядро должно быть лучше и надежнее. Но ряд устройств зависят от модулей, не поставляемых с новым ядром. Особенно это касается графических адаптеров и адаптеров беспроводных сетей. Некоторые дистрибутивы переустанавливают их автоматически; в противном случае потребуются воспользоваться проводной сетью или перезагрузиться со старым ядром и переустановить драйверы для неработающих устройств.

12 Попытка применить pulseaudio

Если вам уже досадила сложность аудиосистем Linux, вы меня понимаете. Если нет, не трудитесь проверять. Ваша звуковая система некоторое время поработает, а потом ни с того ни с сего откажет. Разработчики, пожалуйста, ну дали бы вы нам простую и понятную аудиосистему для Linux – желательно с меньшим числом слоев, чем у лукавицы...

13 Пакеты из другого дистрибутива

Бывает, что в вашем дистрибутиве нет нужного пакета, а в другом дистрибутиве он есть. Не поддавайтесь соблазну установки «чужих» пакетов: это путь к безумию. Может быть, с пакетами Debian в Ubuntu все и обойдется, но рано или поздно (а в случае RPM-пакетов – скорее рано) ждите неприятностей.

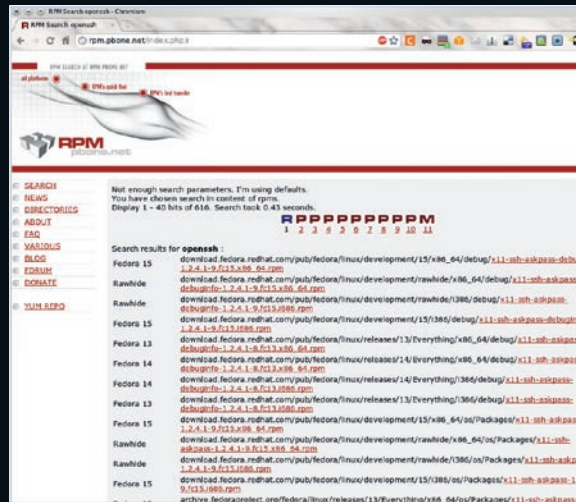
14 Запуск fork-бомбы

Fork-бомба – это мелкий скрипт оболочки, который быстро плодит процесс за процессом, пока компьютер ими не подавится и единственным выходом будет выдернуть его из розетки. Некоторые дистрибутивы предотвращают подобный эффект, ограничивая количество процессов, запускаемых одним пользователем, хотя бы и root. Если кому интересно, что это такое, вот код:

```
#!/bin/sh
:(){ :!:& };
```

15 Переустановка

Предпочтительное «решение» большинства проблем Windows – переустановка ОС. Но проделайте это в Linux, и добьетесь разве что потери всех своих настроек, а также и шансов на выяснение причин проблемы и ее решение. Переустановка отдельных пакетов тоже помогает редко, поскольку не затрагивает настроек в вашем домашнем каталоге, а это наиболее вероятный источник бедствия.



» Если в вашем дистрибутиве нет нужного пакета, не поддавайтесь искушению приспособить пакет из другого дистрибутива – выйдет один вред.

16 Блокировка подкачки

Возможно, вы думаете, что ныне, когда даже в ноутбуках не менее 2–4 ГБ ОЗУ, подкачка стала анахронизмом; но некоторые процессы на нее все же рассчитывают. Даже если вы считаете ее ненужной, память можно переполнить нечаянно, и хотя бы небольшая подкачка не помешает. В конце концов, на современных компьютерах дисковой памяти куда больше, чем ОЗУ.

17 Установка из исходного кода

Если вам нужна новейшая версия малоизвестной (или абсолютно новой) программы, установка из исходного кода бывает единственным возможным вариантом и обычно работает хорошо, но иногда приводит к наличию разных версий одной и той же библиотеки в /usr/lib и /usr/local/lib. Полученные эффекты лаконичнее всего описываются термином «интересно...»

18 Потеря носителя

Linux проники на множество устройств, миниатюрных, удобных, мощных – и легко теряемых. Как ни велик соблазн хранить все подробности своей личной жизни на нетбуке или смартфоне, задумайтесь о последствиях их утраты (с посторонней помощью или без). Само устройство не пострадает, но его новому «владельцу» достанутся все ваши данные, так что будьте осторожны.

19 Обращение как с Windows

И Linux, и Windows – это ОС для настольных ПК, но при внешнем подобию они радикально отличаются. Обращение с Linux как с Windows – одна из самых частых причин проблем начинающих пользователей. Если в Linux вы новичок, не стесняйтесь спрашивать совета у ветеранов, каким бы ни был ваш прежний опыт работы с Windows: наработанные стереотипы здесь могут и помешать.

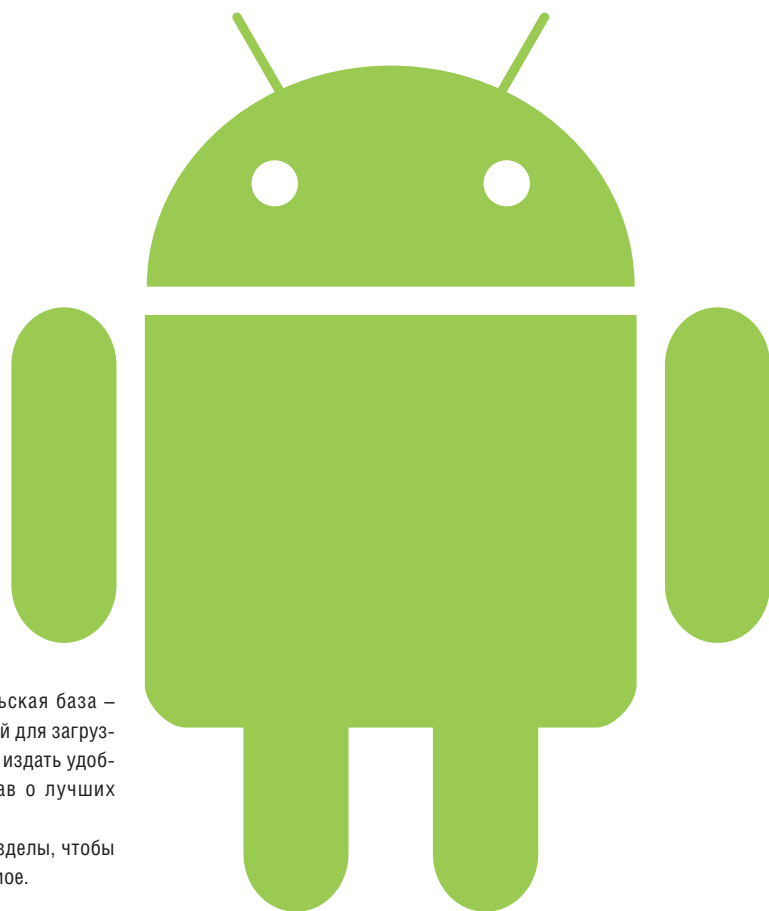
20 Поливание кофе

Достоин сожаления, что система, столь любезная программистам и хакерам, приходит в негодность при контакте с топливом для программистов. Мыши (за исключением живых грызунов) и пицца тоже не очень ладят друг с другом, хотя заядлым линуксоидам, пожалуй, проще обойтись без мыши, чем без пиццы. LXF

Приложения Android

Рынок приложений для Android уверенно растет и уже превысил 200 000 загрузок.

Дэмиен Макферран описывает самые лучшие.



У вас как у фаната Linux наверняка есть смартфон на платформе Android. С момента своего появления в 2008 году революционная мобильная ОС от Google улучшается день ото дня, и сейчас это одна из самых популярных мобильных платформ в мире.

Выросшая пользовательская база – это больше игр и приложений для загрузки, и мы подумали, что пора издать удобное руководство, рассказав о лучших из них.

Мы разделили все на разделы, чтобы вам было проще найти искомое.

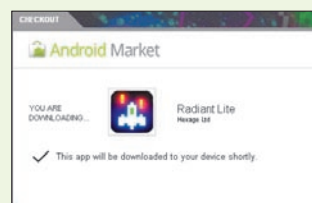
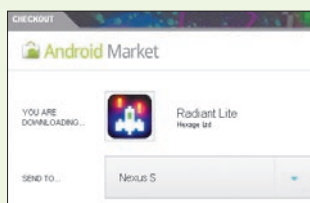
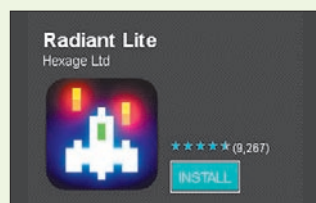
Работа с Android Market на вашем компьютере

ШАГ 1: В Android Market можно зайти с телефона или с настольного компьютера. Вам понадобится учетная запись Google (такая же, которая обычно используется для почты и др.); из нее вы сможете переходить на market.android.com.

ШАГ 2: Приложения распределены по рубрикам; выделены популярные загрузки и загрузки, рекомендованные редакторами Google. Щелкнув на приложении, вы получите подробную информацию о нем, экранные снимки и отзывы пользователей.

ШАГ 3: Выбрав приложение, которое вы собираетесь загрузить, щелкните по ссылке и выберите свое устройство из прилагаемого списка. Если у вас имеется несколько устройств с Android, это тоже можно указать в меню.

ШАГ 4: Выбранное приложение установится на ваш телефон автоматически, «по радио»: вам не придется даже прикасаться к своему телефону или подключать его к чему бы то ни было. И готово! Замечательно, правда?



УТИЛИТЫ

Большинство программ нацелено на что-то одно, и любое достойное загрузки приложение должно и обогатить ваш опыт работы с мобильными устройствами, и слегка упростить вам жизнь. В нашей подборке – программы, без которых не обойдется ни один серьезный пользователь Android.

Barcode Scanner

БЕСПЛАТНО

QR-коды [Quick Response – быстрый отклик] напоминают штрих-коды, и в этой программе можно распознавать QR-коды, снятые на фотокамеру телефона. В результате обычно получаются гиперссылка, которая автоматически открывается в браузере телефона.

VNC Viewer

£4,90

Возможности вашего телефона Android настолько велики, что с его помощью можно даже управлять домашним компьютером – конечно, при посредстве этой программы. Разработанная RealVNC – авторами VNC – эта программа позволяет подключиться к любой рабочей станции с Linux, Mac OS X или Windows, где запущен совместимый VNC-сервер. После входа в систему на экране мобильного телефона появится рабочий стол, и вы сможете выполнять все обычные действия.

AndFTP

БЕСПЛАТНО

Dropbox прекрасен, но этот FTP-клиент позволяет обмениваться файлами между телефоном и домашним компьютером, а также переименовывать и удалять файлы и изменять права доступа к ним. Он может хранить несколько конфигураций подключения, что воплощает мечты занятого работчика.

Dropbox

БЕСПЛАТНО

Сегодня все твердят об облачных вычислениях, но мало кто использует эту технологию на практике. *Dropbox* – один из самых убедительных вариантов, и с его помощью можно без усилий переносить файлы между многими устройствами. Вы просто загружаете файл – фотографию, фильм, песню, что угодно – в свою учетную запись *Dropbox*, и после входа в систему на вашем компьютере (или на любой другой совместимой платформе) файл тут же будет доступен. Процесс работает и в обратном направлении, и одним щелчком мыши можно поделиться файлом с другими.

PhoneUsage

БЕСПЛАТНО (доступна версия Pro)

Большинство людей сегодня пользуются мобильными контрактами с утомительными условиями «добросовестного использования», поэтому загружать музыку и видео с Youtube по 3G, к сожалению, больше нельзя – если только вас не пугает перспектива быть огорошенным огромным счетом. Внимательно следить за загружаемыми данными – дело серьезное, и с помощью *PhoneUsage* выполнять это гораздо проще. Вы сразу увидите объем трафика, за который отвечаете собственным кошелем, и сможете локализовать приложения, бесполезно сжирающие ценные мегабайты.

Better Terminal Emulator Pro

Около £2,40

Как известно, в основе Android лежит Linux, но большая часть полезных свойств открытого ПО похоронена за пользовательскими интерфейсами и настраиваемыми скинами. Если вы чувствуете необходимость копнуть поглубже, можете попробовать эту программу – она эмулирует интерфейс терминала и позволяет запускать родные Linux-приложения и команды терминала с вашего телефона. Без этой невероятно мощной утилиты истинным линуксоидам просто не жить.

Ringdroid

БЕСПЛАТНО

Любителям Apple все еще нельзя сменить рингтоны и системные сообщения на своих iPhone, а у пользователей Android эта возможность была со дня выпуска. В *Ringdroid* можно редактировать любой звуковой файл на карте microSD вашего телефона. Можно подрезать песни, для удобства их применения в качестве рингтонов, или изменить файлы звуковых эффектов при получении текстового сообщения или оповещения о письме.

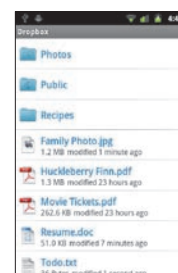
Read it Later Pro

Около £1,80

Как и намекает название, эта программа дает эффективный способ управлять вашим чтением в сети. Адрес поста на форуме, сайта или статьи можно добавить в закладки и вернуться к нему позже. Можно даже сохранить в телефоне всю страницу и на компьютере продолжить чтение с того места, где вы остановились – *Read it Later* прекрасно помнит, где вы находитесь.



➤ Сканированные QR-коды сразу станут гиперссылками.



➤ Облачное хранение данных в умном Dropbox.



➤ Отредактируйте свои рингтоны с Ringdroid.

Игры

Стех пор, как мы упражняли терпение на Java-играх «тупых телефонов» Nokia и Sony Ericsson, мобильные игры прошли большой путь. Многие игры на Android способны потягаться с играми для Nintendo DS и Sony PlayStation и предоставляют отличный способ потратить минутку, а то и десять.

Angry Birds

БЕСПЛАТНО

Angry Birds прославились в мире, конкурируя с *Tetris* на Game Boy. Доступная во множестве сборок для Android, эта с виду простая головоломка – видеоигра, напоминающая лист пузырчатой пленки – очень заразительна, и перестать играть прямо нет сил. Очень убедительный пример зрелости игр для сенсорного экрана.

Fruit Ninja

Около 80 пенсов

Цель игры – нарезать фрукты пальцем, избегая контакта с бомбами. Бонусы начисляются за разрезание нескольких фруктов одним взмахом, а возможность разблокировать всякие секретные достижения заставляет вас возвращаться к игре снова и снова. 3D-графика очень приятна на вид, и вы никогда не устанете от забавного

хлюпающего звука, сопровождающего каждый точный удар.

Sacred Odyssey: Rise of Ayden

£3 (только на сайте Gameloft)

После нескольких минут игры в *Sacred Odyssey* становится понятно, что вдохновляло эту эпическую приключенческую игру. Серия *Zelda*, бестселлер Nintendo, кажется, питал все аспекты этой трехмерной RPG, и хотя такое подражание, безусловно, разочарует игроманов-пуристов, при столь высоком качестве результата обижаться особо не приходится.



➤ Fruit Ninja: так просто и так увлекательно.

Эмуляторы



Доля рынка приложений Android, которая часто игнорируется – эмуляция старых игровых приставок. «Сырая вычислительная мощность», содержащаяся в большинстве современных телефонов, позволяет им без особых усилий воспроизводить функционирование почти любой игровой приставки за последние 30 лет, по сути превращая телефон в портативный зал игровых автоматов.

FPse for Android

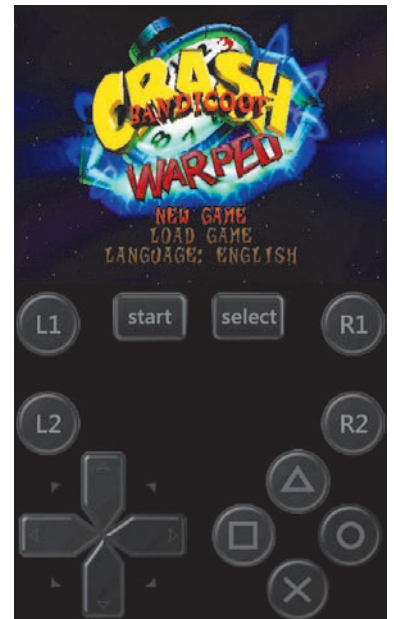
Около £3,15

Если бы в середине девяностых вы сказали нам, что однажды мы сможем играть в *Gran Turismo* на мобильниках, мы бы ответили, что вас серьезно надули и, возможно, вы даже слегка не в себе. Однако теперь у нас есть *FPse* – эмулятор 32-битной PlayStation, позволяющий тщательно воссоздать любую из игр для популярнейшей игровой консоли Sony в телефоне, лежащем у вас на ладони. Его производительность зависит от быстродействия и мощности телефона (с процессором менее 1 ГГц не стоит и беспокоиться), но если ваш телефон соответствует требованиям, вы получите поистине незабываемый опыт.

SNES9X EX

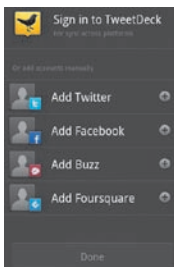
БЕСПЛАТНО

16-битная SNES от Nintendo считается одной из самых лучших когда-либо созданных игровых консолей, с такой винтажной классикой как *Super Mario World*, *Street Fighter II*, *Zelda: Link to the Past* и *Super Mario Kart*. *SNES9X EX* позволяет насладиться их творческим наследием в устройстве Android на полной скорости и предусматривает бездну настроек для дальнейшего улучшения производительности. Приложение способно даже подключаться через Bluetooth к удаленному контроллеру Wii Nintendo, и настоящие кнопки сделают ваши ощущения реалистичными.

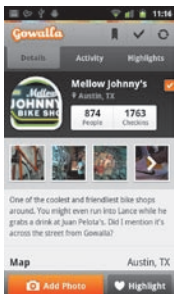


› *Crash Bandicoot* на вашем мобильнике? Невероятно, но факт.

Социальные сети



› Социальных сетей много – лента одна.



› В отъезде ваша связь с друзьями не прервется.



Сегодня все делают это, и стоит ли вам отставать? Социальные сети, похоже, проникли во все сферы современной жизни, и эта подборка приложений поможет вам делиться контентом и оставаться в контакте с быстро растущим кругом цифровых друзей.

TweetDeck

БЕСПЛАТНО

Следить за Twitter, FourSquare, Google Buzz и Facebook бывает утомительно даже в лучшие времена, но, к счастью, великодушный *TweetDeck* даст вам некоторую передышку. Он сливает обновления лент социальных сетей в один поток, и вам не придется запускать несколько приложений одновременно. Более того, он позволяет добавлять записи сразу в несколько каналов. Весь этот функционал обернут в симпатичный и стабильный интерфейс.

AppsFire

БЕСПЛАТНО

Заявленная как лучший способ разведывания новых приложений, *AppsFire* – утилита социальной сети – позволяет наблюдать за загрузками других пользователей Android. Доступны всякие списки – популярных приложений, приложений со скидкой и приложений, рекомендуемых «VIP-пользователями» *AppsFire*. Можно показать другим, какие программы стоят у вас, или отправить ссылки на конкретные загрузки друзьям. Для новичка в мире Android это лучшая стартовая площадка.

Gowalla

БЕСПЛАТНО

Социальные сети на базе местоположения сейчас в фаворе, во многом благодаря технологиям GPS, реализованным в большинстве современных смартфонов. Идея «регистрации» в местах, посещаемых вами ежедневно, может показаться видеогрой в виртуальной реальности для психов, но это на удивление увлекательный опыт, и так проще следить за своими друзьями и знакомыми и организовывать импровизированные встречи. *Gowalla* умеет строить графики ваших перемещений, загружать фотки и даже размещать сообщения в сетях типа Twitter и Facebook; это идеальный компаньон для путешественника и – что, пожалуй, важнее – любителя рассказывать о своих вояжах другим.

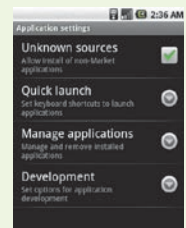
Сторонние приложения для Android

В приложениях Android прекрасно то, что при их установке вы не ограничены одним Android Market. Файлы APK можно устанавливать вручную, но позаботьтесь снять галочку «Unknown Sources [Неизвестные источники]» в меню «Applications Settings [Настройки приложения]» своего телефона.

После этого через любой файловый менеджер (например, *Ol File Manager*) найдите файл APK и установите его. Следует помнить, что сторонние приложения требуют осторожного отношения, особенно если вы не уверены в их происхождении. Хотя Android Installer предупредит вас о том, какие сер-

висы телефона – такие как доступ к сети – нужны для функционирования приложения, существуют приложения, которые могут способствовать хищению вашей личной информации.

› Устанавливать приложения вручную можно, если осторожно.



Украшители телефона

Средний телефон с Android – по сути, мини-компьютер, но место для улучшения есть всегда. С помощью этой подборки незаменимых программ вы превратите свой смартфон в «суперфон».

Tasker

£3,99

С этой безумно мощной утилитой автоматизации можно управлять функционалом своего телефона практически на любом уровне. Надоело включать и отключать Wi-Fi? Создайте в *Tasker* правило, которое предпишет вашему телефону включать его дома и в пределах личной сети. Аналогично можно создать событие, которое автоматически приглушит телефонный звонок в ночное время. Программирование правил требует времени и усилий, но полученные результаты более чем стоят этих усилий.

FlexT9

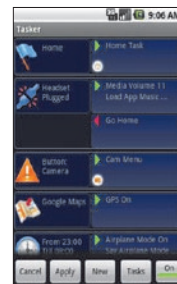
Около £3

Печать «тук-тук» может сравниться со старомодным прогнозирующим текстовым интерфейсом T9, но по скорости им далеко до методов trace-to-type. *Swipe* считается одним из лучших вариантов замены клавиатуры в Android, но пока он находится в состоянии закрытого бета-релиза и доступен, как правило, только предустановленный. *FlexT9* – жизнеспособная альтернатива, заодно предоставляющая распознавание голоса и поддержку почерка. Осознав, как это здорово, вернуться назад будет трудно.

OI File Manager

БЕСПЛАТНО

Как ни странно, в большинстве телефонов Android нет файлового менеджера, но многие разработчики написали свои, и они сильно экономят усилия. В интуитивно понятном интерфейсе можно перемещать файлы, очищать мусор и даже устанавливать приложения и загружать файлы прямо в свою учетную запись *Dropbox*.



➤ *Tasker* превратит ваш скромный мобильник в полнофункциональный смартфон.

Screebi

Около £1,20

Экран телефона так и потребляет энергию, и надо строго задать период отключения подсветки. Но тогда придется все время трогать экран, чтобы она не гасла. *Screebi* видит по датчику движения телефона, что тот у вас в руке, и подавляет отключение.

Покупки

С этими специализированными приложениями совершать покупки онлайн стало еще проще. Вам больше не придется выходить из дома ради шоппинга – делайте все это, сидя в кресле с телефоном в руках.

eBay

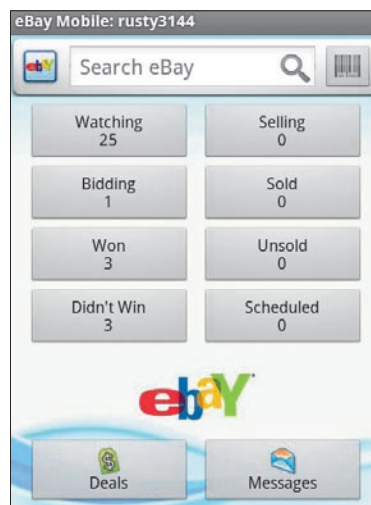
БЕСПЛАТНО

У всеми любимого аукциона есть свое официальное приложение для Android, и оно, естественно, незаменимое. С телефона можно выполнять поиск в английском eBay (и в других странах), а зайдя на сайт – просмотреть список отслеживаемых товаров, сделать ставку и приобрести товары с возможностью мгновенной покупки ['Buy it Now'], и даже отправлять и получать сообщения. Благодаря тесной интеграции с PayPal можно искать товары, делать ставки и оплачивать покупки, не выходя из приложения.

Amazon UK

БЕСПЛАТНО

Amazon становится местом номер один для онлайн-покупателей, и из его официального приложения просмотреть огромный список товаров продавца просто как никогда. Здесь есть все, что есть в Сети, и поддержка заказа одним щелчком делает процесс покупки на сто процентов безопасным, так как вам не придется вводить свой адрес или данные кредитной карты. Можно даже сфотографировать штрих-код товара камерой телефона и проверить, нет ли этого товара в базе данных Amazon по более дешевой цене.



➤ Не упустите сделку.



Замена пускателя приложений

В отличие от телефонов конкурентов, в Android можно заменить домашний экран по умолчанию, известный также как «пускатель приложений» [Launcher]. Если вы хотите придать своему телефону на Android свежий вид, вот что вам необходимо...

LauncherPro

БЕСПЛАТНО (доступна версия Pro)

Назначьте число домашних экранов, задайте клавиши быстрого доступа для прокручиваемой панели и к выпадающим меню. *LauncherPro* также может похвастаться эксклюзивными виджетами и набором анимаций перехода между экранами.

ADW Launcher EX

Около £2

Главный соперник *LauncherPro*, *ADW Launcher* предоставляет похожие функции и не менее привлекателен. Его козырь – наличие пользовательских тем: они изменяют вид домашних экранов и содержат специально разработанные иконки.

SPB Shell 3D

Около £9

SPB Shell 3D задаст встряску графике вашей трубки. Вас ожидают кардинальная реформа всей системы меню, а также потрясающие 3D-эффекты и анимация, которые придадут вашему телефону высший шик. Программа еще не вполне совершенна – она изрядно нагружает процессор, а ее цена немногим ниже десяти фунтов; но отрицать, что это потрясающая замена для пускателя приложений, невозможно. **LXF**



➤ *LauncherPro*: и как вы раньше-то без него обходились?

Зарождение Linux: Причины и предпосылки

Алексей Федорчук продолжает серию статей об истории наиболее успешного OpenSource-проекта.

Пока, как было сказано в предыдущей статье, в Беркли и его окрестностях поневоле занимались юридическим крючкотворством, на другом конце света, в Финляндии, некий студент по имени Линус Торвалдс [Linus Torvalds] размышлял, что же ему делать с только что приобретенным IBM PC. И, как ни странно, результаты его размышлений оказали не меньшее влияние на нашу историю, нежели многолетний труд исследователей, финансируемых правительством мировой державы.

Впрочем, рождественскую сказку о том, как бедный студент копил деньги на 32-битный компьютер, а потом сочинял программу терминального доступа к удаленной университетской машине, которая затем превратилась в полноценную ОС, не описывал только ленивый. Известна она и в версии от создателя этой ОС – самого Линуса Торвалдса. Так что пересказывать ее в очередной раз я не буду. А попробую, в меру своего понимания, выявить причины и предпосылки появления Linux. Правда, к вопросу о бедном студенте нам придется еще вернуться.

Причины

Главная причина появления Linux – «железная». С начала 80-х годов по всему миру шло триумфальное шествие Советской вла-

сти... то есть, pardon, Intel-совместимых PC. Первый широко распространенный персональный компьютер, собственно IBM PC, базируясь на внутренне 16-разрядном процессоре Intel 8088, работать под изначально 32-битной UNIX не мог. Как не способна была на это и персоналка следующего поколения, IBM PC/AT на процессоре Intel 80286.

Однако появление в 1985 году первого 32-разрядного процессора от Intel, 80386, дало возможность использовать UNIX на дешевых и общедоступных персоналках. А появление в 1989 году процессора Intel 80486 вплотную приблизило их производительность к вычислительной мощности рабочих станций на RISC-процессорах. Росли объемы оперативной памяти, емкость жестких дисков, совершенствовалась видеосистема, появлялась первая мультимедийная периферия. А самое главное – PC становились все многочисленней, стандартизованней и дешевле; против этого триединства факторов все иные архитектуры массовых платформ общего назначения оказались бессильны.

Но под чем же работало все это аппаратное богатство? Да в подавляющем большинстве – под MS DOS, 16-разрядной операционной системой, созданной еще для первых IBM PC и несущей в себе массу неустраняемых ограничений: принципиальную однозадачность, отсутствие многопользовательского доступа, возможность использовать «по прямому назначению» лишь 640 Кбайт оперативной памяти, примитивную организацию файловой системы, не менее примитивные средства работы в текстовом режиме – единственно возможном силами «черного» DOS.

Конечно, предпринимались многочисленные попытки заретушировать «родимые пятна» DOS. Разрабатывались надстройки над ней, способные использовать весь физический объем оперативной памяти и многозадачность – такие как QuaterDesq и Geoworks, которые включали также и системы работы в графическом режиме. Некоторые пользовательские DOS-приложения (табличные процессоры Lotus 1-2-3 и QuattroPro, текстовый редактор WordPerfect) обзаводились собственными средствами управления памятью и графическими интерфейсами.

Вся эта многочисленная DOS-косметика была либо неудачной, либо не получила распространения. Конечно, существовала и альтернатива ей – разрабатывавшаяся в IBM операционка OS/2, первая 32-разрядная ОС, специально написанная для PC. Однако и она, несмотря на весьма прогрессивный базис, не приобрела широкой популярности.

Сложилась парадоксальная ситуация: «народная», то есть общедоступная, платформа не имела адекватной «народной» же



Compaq DeskPro 386 – первая 32-битная персоналка, пригодная для запуска UNIX-подобных операционных систем.

операционки, способной использовать ее возможности. В конце предыдущей статьи мы, казалось бы, обозначили претендента на роль народной ОС для современных, на тот исторический момент (начало 90-х годов прошлого века), компьютеров – FreeBSD.

Однако судьба распорядилась иначе. И «святое место» массовой операционки для настольных персоналок оставалось пусто вплоть до 1995 года – появления Windows 95. Началась эра гегемонии платформы Wintel (то есть машин на Intel-совместимых процессорах под управлением ОС Windows).

Гегемония эта практически не поколеблена и по сей день. Однако уже на протяжении многих лет она имеет альтернативу. Вследствие описанного в прошлой статье «технологического» детектива ею стала не FreeBSD, а совсем другая система.

Прежде чем переходить к предпосылкам ее возникновения, сделаю маленькое отступление. Параллельно с развитием платформы Wintel существовали и другие варианты, на иной аппаратной базе – процессоре Motorola 680x0, Macintosh и Amiga, с их System # и AmigaOS, соответственно. Однако обе они занимали нишевое положение – не в последнюю очередь, вследствие цены конечных решений. О чем мы сейчас и поговорим.

Предпосылка первая: цены

Как ни печально русским интеллигентам это признавать, но первой предпосылкой для возникновения Linux была именно цена. Потому что именно в начале 90-х годов машины с 32-битными процессорами от Intel (а затем от AMD и Cyrix) стали доступны народу.

Причины этого явления уходят еще в 80-е годы. Когда IBM выпустила свой первый PC, еще 16-битный, руководство фирмы не относилось к этой затее особенно серьезно и не рассчитывало «нарубить капусты» на этом рынке. В результате все спецификации архитектуры были открыты, и не имелось ни технических, ни юридических препятствий к клонированию этих машин.

Чем немедленно воспользовались многие производители, сначала американские, а затем европейские и восточноазиатские. И очень скоро число клонов (так называемых IBM-совместимых машин) превысило число оригиналов – IBM PC-XT и PC-AT. А потом была выпущена (в 1986 году) и первая машина на 32-битном процессоре от Intel (80386). И выпустила ее вовсе не IBM, а фирма Compaq – она получила имя Compaq DeskPro 386.

Тут в IBM спохватились, что рынок, созданный их стараниями, от них уходит – и разработали новую архитектуру – PS/2, также на процессоре x86, но с закрытыми спецификациями, исключающими клонирование. Однако было поздно: рынок был заполнен стандартными PC. К рубежу 80-х и 90-х годов их не производил только ленивый – начиная от компьютерных гигантов типа Hewlett-Packard и заканчивая дядюшкой Мяо с Малой Арнаутской улицы острова Тайваня. К которому вскоре присоединился его родич – дядюшка Ляо с ее продолжения на континентальном Китае. И ныне эти два дядюшки обеспечивают мир львиной долей комплектующих для компьютеров – и даже готовыми системами, под какими бы торговыми марками последние не продавались.

Результатом был лавинообразный обвал цен. Первый IBM PC-XT стоил более трех тысяч долларов, а для полноценной работы требовал еще и доукомплектования на треть этой суммы. Ко второй половине 80-х годов относятся слова Линуса Торвальдса: «Раньше было такое правило, что все компьютеры начального уровня стоили две тысячи». В первые же 90-е годы порог вхождения в мир IT снизился до тысячи баксов. То есть компьютеры не скатились еще до цены ширпотреба, как ныне. Но все, кому та-



➤ Эндрю Таненбаум – создатель «игрушечной» ОС MINIX.

кие машины нужны были «по делу», были в силах приобрести их, причем не только «у них», но и «у нас».

Предпосылка вторая: ОС

В числе тех, кому такие машины были действительно нужны, и кто мог, тем или иным путем, ими обзавестись, оказались и студенты, обучавшиеся по специальностям, связанным с IT-сферой. Но вот с операционной системой для 32-битных компьютеров у студентов оказалось хуже: FreeBSD еще не начала свое свободное плавание, а коммерческие UNIX'ы стоили несколько дороже, чем необходимое для их работы «железо». Да и не густо было с ними: на архитектуре i386 были способны работать только SCO UNIX и Xenix – кастрированный UNIX от Microsoft.

И тут в канву нашего сюжета вписывается «игрушечная» ОС MINIX. Ее явил миру в январе 1987 года Эндрю Таненбаум [Andrew Stuart Tanenbaum], профессор Университета Вреие, Амстердам, Нидерланды. И преподавал он в этом университете не что иное, как Computer Science, хотя и был по образованию физиком [стоит отметить, что название университета – Vrije – по-

голландски означает «свободный», – прим. ред.].

Университетское образование в области компьютерных наук в 80-е годы прошлого века базировалось преимущественно, если не исключительно,

на UNIX. Что, как явствует из сказанного ранее, создавало для студентов известные трудности.

Так вот, Таненбаум вел в вышеозначенном университете курс UNIX, к которому написал собственный учебник – «Operating Systems: Design and Implementation». Но изучать UNIX без системы – все равно, что обучаться музыке без инструмента. А с инструментом-то как раз и была напряженка. И ему не осталось ничего другого, как такой инструмент изготовить. Им-то и стала ОС MINIX (в дальнейшем получившая имя MINIX 1), вышедшая в свет в 1987 году.

«Машины с 32-битными процессорами стали доступны народу.»



➤ **Линус Торвальдс — он не только ядро пишет.**

Это была маленькая и компактная операционка, работавшая на машинах с архитектурой i386. Доступность MINIX увеличивалась еще и тем, что ее можно было скомпилировать даже в 16-битном варианте, и в этом качестве она становилась пригодной к использованию не только на PC-AT (80286), но даже, как говорят, на XT'шках, то есть на машинах с процессором 8086/8088.

Распространялась она исключительно как сопроводительный материал к упомянутому выше учебнику. Весь комплект, по свидетельству Линуса Торвальдса, стоил 169 долларов при заказе по почте. Что на самом деле не так дорого: в те годы на Западе, только-только переставшем загнивать, ни одно специализированное книжное издание не стоило дешевле 100 баксов. Так что фактически основная, если не вся, затратная часть для пользователя приходилась на книжку, да и дискеты были не так дешевы. Сама же ОС как таковая могла рассматриваться в качестве бесплатного приложения к книге и носителям. И, во всяком случае, это было несоизмеримо дешевле тех тысяч долларов, в которые обходилась лицензия на любой из существовавших тогда проприетарных UNIX'ов. Требовавших, к тому же, сущей бездельницы в виде соответствующей рабочей станции за несколько десятков тысяч.

Разумеется, ОС MINIX распространялась в сопровождении исходных текстов, предназначенных для изучения и потрошения — необходимость в котором возникла очень скоро.

Дело в том, что ОС MINIX, предназначенная исключительно для учебных целей, в принципе не была приспособлена для выполнения каких-либо реальных задач. Однако шаловливые студенческие (и не только) ручки так и чесались прикрутить ее к чему-либо пригодному для практического использования. В результате система очень быстро обросла всякого рода заплатками, из которых главным был патч от австралийца Брюса Эванса. После нало-

жения этих заплат система становилась способной выступать как платформа разработчика. Именно на такой заплатанной системе Линус Торвальдс спустя несколько лет начнет создавать свою операционную систему.

Однако сама по себе MINIX по-прежнему распространялась исключительно в первозданном виде — как чисто учебная система, и лишь в сопровождении книги (или, напротив, сопровождая книгу). То есть, будучи открытой, она не была свободной. Ибо права на MINIX принадлежали издательству Prentice Hall, выпустившему учебник Таненбаума. В сущности, правовой статус MINIX был точно таким же, как и обычной книги. Что, однако, не мешало тому, что на протяжении десяти, а то и более, лет по ней учились поколения студентов — как до Торвальдса, так и после него.

Надо заметить, что Таненбаума нельзя рассматривать только как предтечу Линуса, а его систему — как трамплин для его разработки. Кроме упомянутой выше «Operating Systems: Design and Implementation» (в переводе: «Операционные системы: разработка и реализация»), его перу принадлежат:

- » «Computer Networks» («Компьютерные сети»);
- » «Modern Operating Systems» («Современные операционные системы»);
- » «Structured Computer Organization» («Архитектура компьютера»);
- » «Distributed Systems: Principles and Paradigms» («Распределенные системы. Принципы и парадигмы»).

Все они по праву принадлежат к классике жанра IT-литературы, выдержали по несколько изданий (Computer Networks и Structured Computer Organization — аж по пять) и переведены на многие языки. В том числе и на русский: они выпущены издательством «Питер» в серии «Классика Computer Science».

Главной разработке Таненбаума, MINIX, судьба также уготовила вторую жизнь. Долгое время она продолжала эволюционное развитие в качестве учебной системы — были выпущены версии MINIX 1.5 (1992 год) и MINIX 2 (1997 год), представлявшие собой «песочницы» для начинающих юникоидов. Однако кардинал лелеял коварные замыслы: превратить MINIX в полноценную операционную систему, реализующие его представления о том, какой должна быть современная ОС. А заодно — сделать ее свободной в полном понимании этого слова: ведь «несвобода» предыдущих версий объяснялась не жадностью профессора, а спецификой издания и распространения.

Результатом явился анонс новой операционки, MINIX 3, который состоялся 24 октября 2005 года. Это была не просто следующая по номеру версия,

а именно новая операционная система, почему цифру «3» здесь следует рассматривать как часть ее имени собственного. Обособленность от предшественников подчеркивалась и тем, что отныне MINIX 3 будет распространяться не как довесок к книге, а совершенно самостоятельно, под лицензией BSD.

С трудами Эндры Таненбаума, которые не ограничиваются перечисленными выше (достаточно вспомнить проекты Amoeba и Globe) можно ознакомиться на его личной странице: <http://www.cs.vu.nl/~ast/>.

Главный герой выходит на сцену

Итак, толчком для написания Линусом собственного ядра послужила MINIX — «студенческая» операционка Энди Таненбаума, с помощью заплат приспособленная для выполнения практической работы.

Однако сам Линус не занимался «доведением MINIX до ума». Не использовал он также и код какой-либо из реализаций UNIX или BSD. Он воссоздал функциональность ядра UNIX с нуля — руководствуясь описаниями системных вызовов, данными в со-

ответствующем стандарте POSIX. И потому Linux не является ни клоном System V, ни клоном BSD – хотя в ней и использована схема инициализации в стиле первой, да и идейное влияние второй, безусловно, имело место быть.

Здесь надо сделать отступление и рассказать об истории стандарта POSIX. Для чего придется вернуться назад, ко временам, когда UNIX, выйдя за пределы породившей его AT&T, начала интенсивно делиться на ветки – как проприетарные, так и те, которые со временем станут свободными. В основе этих веток лежали разные базовые системы – SVR3, SVR4, 4BSD, и каждый разработчик считал своим долгом внести те или иные усовершенствования. А поскольку все существовавшие тогда системы, кроме BSD, были закрытыми, усовершенствования эти плохо согласовывались между собой.

Это ставило под угрозу один из краеугольных камней UNIX-идеологии – портируемость приложений. И начался процесс, который, вслед за товарищем Мао, можно назвать «борьбой за упорядочивание трех стилей работы». Однако председателю КПК было попроще – в данном случае речь шла не о трех, а едва ли не о тридцати трех стилях.

Тем не менее, процесс упорядочивания пошел, реализуясь в виде ряда стандартов, наибольшее признание из которых получил POSIX – Portable Operation System Interface based on UNIX, разработанный международной организацией IEEE (Institute of Electrical and Electronics Engineers, Inc.). И любая операционная система, претендующая на звание UNIX-совместимой, должна этому стандарту соответствовать.

Крестным отцом термина POSIX стал Ричард Столлмен. Слово Portable в его названии первоначально означало, что соответствующая POSIX-спецификациям система может быть перенесена на любое компьютерное «железо». Однако со временем не менее важным оказался несколько другой аспект этого термина: любая прикладная программа, написанная в соответствии со стандартами POSIX, может быть перенесена/портирована на любую совместимую с ним ОС.

Стандарты POSIX были приняты в 1988 году и зафиксированы в виде серии регулярно обновляемых документов (общим числом под два десятка), в которых описываются спецификации отдельных компонентов системы.

Именно на эти документы в первую очередь и опирался Линус Торвалдс, создавая свою ОС по мотивам MINIX.

Linux создавался на машине с процессором i386 для архитектуры Intel, и первоначально – только для нее. Более того, долгое время Линус вообще сомневался, что его система когда-либо сможет быть портирована на любую иную аппаратную платформу. И потому соответствие стандартам в данном случае преследовало целью не переносимость Linux самого по себе, а в первую очередь возможность компиляции в этой ОС всего ранее созданного программного ассортимента для UNIX и POSIX-совместимых систем вообще.

Лично Линусу принадлежит честь разработки ядра Linux и файловой системы ext (то есть Extended – расширение для файловой системы Minix), как и сменившей ее впоследствии ext2. В качестве среды для работы он выбрал bash – командную оболочку, разрабатываемую в рамках проекта GNU. А для сборки своего кода использовал тот самый GCC, о котором говорилось в предыдущей статье; главной же общесистемной библиотекой функций языка Си выступала ее GNU-реализация, *glibc*. Все прочее системное окружение ядра – комплекс пакетов, который можно назвать Base Linux – также в основном происходит из проекта GNU. Да и при выборе политики распространения

Линус в конце концов остановился на лицензии GPL – детище Ричарда Столлмена и его Фонда свободного программного обеспечения (FSF).

GNU ли Linux?

На основании сказанного выше часто полагают, что ОС Linux должна на самом деле именоваться GNU/Linux. Правильно ли это?

По моему скромному мнению – нет. Конечно, роль программного обеспечения, разработанного в рамках проекта GNU, для развития Linux как пользовательской платформы переоценить трудно. Однако не проект GNU ухватился за столь недостающее ему ядро. Напротив, это Линус для обеспечения работы своего ядра использовал отдельные компоненты из GNU-арсенала. В полном, к слову сказать, соответствии с духом и буквой GPL и движения FSF. Впрочем, те, кто считает нужным подчеркнуть роль компонентов GNU в составе Linux, вполне могут это делать – и делают.

Добавлю еще, что неотъемлемой чертой Base Linux является альтернативность его комплектации. И потому ОС Linux – не только (а может быть, и не столько) ядро и набор базовых программ,

но в первую очередь алгоритм для построения такого набора. И создание такого алгоритма – второе, после написания кода ядра, великое достижение Линуса.

Наконец, Линус оказался создателем уникального метода разработки масштабных проектов Open Source, того самого, который Эрик Реймонд позднее назовет методом большого базара. Впрочем, справедливости ради следует отметить, что в данном случае и он изобрел велосипед – аналогичный способ привлечения дармовой рабочей силы использовал Том Сойер в своих «Приключениях». Однако, если инструментами Тома были сердцевина от яблока и крыса с привязанной к хвосту веревкой, чтобы удобнее размахивать ей над головой, то орудием Линуса оказался Интернет.

Заключение

А теперь я опять вернусь к теме предыдущей статьи. Существует мнение, что если бы BSD (еще не разделившаяся на Net- и FreeBSD) не погрязла бы в тяжбе с AT&T и получила бы свободу в конце 80-х – начале 90-х годов, то в разработке Linux не было бы никакой необходимости. Несмотря на свою пылкую любовь к BSD-системам во всех их проявлениях, не могу с этим согласиться: если бы Linux'a не было – его следовало бы изобрести. Потому что без него жить было бы скучно...

Самое же главное, что камерный, во многом академический, стиль разработки BSD-систем, при всех его достоинствах с точки зрения качества, не способствовал вовлечению в этот процесс широких народных масс. И, как следствие, не мог был бы обеспечить открытому и свободному софту той популярности, которую он приобрел благодаря Linux.

Собственно, рождение Linux дало толчок к окончательному оформлению движения Open Source, несколько обособившемуся от сообщества Free Software – хотя и по сей день это существенно пересекающиеся множества. Но если апологеты FSF, во главе с Ричардом Столлменом, декларируют, что все программное обеспечение должно быть свободным, исходя из моральных и идеологических соображений, то для сторонников Open Source характерен более прагматический подход. Их принцип – открытое программное обеспечение следует использовать не потому, что оно открытое, свободное или бесплатное. А потому, что оно просто лучше проприетарного. В том числе – и вследствие публичной экспертизы, реализуемой именно благодаря внедренному Линусом методу Тома Сойера. **LXF**

«Линус оказался создателем уникального метода разработки.»

Что за штука... coreboot?

Думаете, ваш компьютер вполне свободен? BIOS-то наверняка проприетарная. Джонатан Робертс вникает в проект, намеренный исправить дело.

» Coreboot, говорите? Это про загрузку компьютера?

Вы совершенно правы. В частности, coreboot должен заменить устаревшие проприетарные BIOS, работающие на большинстве ПК, на современную альтернативу с открытым исходным кодом.

» Минутку, а что такое BIOS?

Аббревиатура разворачивается так: Basic Input/Output System (то есть «базовая система ввода-вывода»). Это крохотная программа, которая хранится на микросхеме и вступает в работу

«К BIOS есть по меньшей мере три серьезных вопроса.»

сразу при включении питания. Система знакома каждому, кто хоть раз загружал компьютер: обычно это первый экран после «представления» видеокарты, как правило, с логотипом производителя.

» Да уж, мне всегда кажется, что он там год маячит...

Ну, BIOS выполняет много работы, а на это требуется время.

Во-первых, базовая система выполняет несколько тестов: в частности, выясняет объем доступной памяти, инициализирует и регистрирует доступные устройства, а затем формирует среду для работы операционной системы.

Выполнив базовую настройку, BIOS определяет загрузочное устройство и передает управление ему.

» Как все сложно! Но если эта штука сидит на большинстве существующих ПК, значит, работает хорошо? Зачем тогда ее заменять?

Действительно, работает, но к BIOS есть по меньшей мере три серьезных вопроса, требующие безотлагательного рассмотрения.

» Каковы же эти вопросы?

Начнем с проблемы дублирования усилий. Большая часть инициализации устройств, выполняемой BIOS, была включена для поддержки устаревших операционных систем типа DOS. В них не хватало драйверов, и BIOS инициализировала для них оборудование и служила связующим звеном.

» Но ведь в современных ОС драйверы есть?

Да. Операционные системы типа Windows, Linux и Mac OS поставляются с собственными драйверами. В результате по окончании работы BIOS весь процесс повторяется заново.

Эффективности тут явно мало, и поскольку современные ОС гораздо лучше поддерживают аппаратуру и загружают драйвера

с большим умом, имеет смысл передать эту задачу на уровень операционной системы.

» Выходит, это из-за BIOS мой компьютер тормозит?

Да. Для рядового пользователя это вроде и не вопрос – но представьте себе компьютерный кластер, где на каждую систему приходится по 128 ГБ ОЗУ и десяток сетевых портов. Здесь тестирование аппаратуры в BIOS займет немало времени.

» Ну, здесь-то не помешало бы исправить положение, но ситуация все же не выглядит безнадежной. Что еще можно улучшить?

Как я уже упоминал, BIOS большинства ПК очень стары. Поэтому сегодня практически никто не знаком с их внутренней «кухней» и диагностическими средствами (кроме, конечно, Майка: его собственная ОС постоянно общается с BIOS).

» Ой, так если у меня слетит BIOS, мне и подсказки никто не даст?

Вот именно, и лишь немногие вообще знают, с чего тут начать. То же самое произойдет, если понадобится расширить поддержку BIOS на какое-нибудь

экзотическое устройство. BIOS написаны на языке ассемблера 8086, и расширения тоже надо писать на этом языке. Майку-то хорошо, но у большинства современных программистов в ассемблере опыта мало. Так что добавление поддержки нового оборудования может обернуться кошмаром.

» Да, причина серьезная. А в чем последняя проблема?

Вероятно, последняя проблема серьезнее всех, но это зависит от вашей параноидальности. Проприетарная BIOS, которой комплектуется большинство современных систем, по определению получает неограниченный контроль над оборудованием компьютера в процессе загрузки.

Логично предположить, что, сдав дела операционной системе, BIOS уходит на покой. Как бы не так! BIOS продолжает работать, то есть сохраняет доступ к дискам и хранящимся на них данным и ко всем сетевым подключениям. Неконтролируемый двоичный код, написанный неизвестно кем – для любителей безопасности это страшно.

» Ужас! И вот этот самый coreboot все исправит? Но как?

Да, исправит. Сперва планировали просто залить Linux на микросхему – туда, где обычно гнездится BIOS. Это устранило бы все названные проблемы: ведь Linux знакома многим людям, драйверы грузились бы только раз, а проверить открытый код мог бы каждый.

Поскольку планировалось заменить BIOS на Linux, проект назвали LinuxBIOS.

» «Сперва» планировали? Выходит, случился облом?

Увы... Хотя Linux могла сделать значительную часть работы (загрузить драйверы и т.п.), оказалось, что устройства, еще не подключенные к электропитанию, она считает аннулированными.

К тому же выяснилось, что на некоторых системах Linux не в состоянии готовить к работе многоядерные процессоры и память. Поэтому команде coreboot пришлось писать специальный код,



Замена для BIOS,
уже с открытым
исходным кодом

coreboot

Поддерживается
свыше 230 видов
системных плат

Быстрее,
безопаснее,
универсальнее
и понятнее

чтобы он настраивал аппаратуру на самом низшем уровне, а затем передавал управление Linux.

» Ага, я понял... этот код составляет ядро [core] процесса загрузки [boot], от которого танцует остальное. Вот его и назвали coreboot. Совершенно точно.

» Но если пришлось переписывать код, не пропадут ли преимущества, которые планировались благодаря LinuxBIOS?

Не думаю. Начнем с того, что coreboot почти целиком написан на C и почти не содержит ассемблера. Язык C весьма популярен, поэтому модифицировать и совершенствовать код сможет масса программистов. Кроме того, исходный код coreboot открытый, и бесконтрольных участков двоичной абракадабры в системе уже не будет.

А коли работа по подготовке среды для Linux проводится минимальная, не будет и дублирования усилий.

Данная концепция неожиданно принесла и ряд побочных выгод.

» Это каких же?

Закончив работу, coreboot попросту запускает «полезную нагрузку». Полезная нагрузка – это другой экземпляр программного обеспечения, хранящийся на одном чипе с coreboot и выполняющий особую функцию.

Таковым может быть ядро Linux (хотя объем современных чипов пока не дает такой возможности). А может быть и загрузчик типа *Grub* или *SeaBIOS*. Такой загрузчик вызовет хоть Windows, хоть Linux – любую операционную систему – с жесткого диска. Или это может быть одна из слу-

жебных программ, или игра... список можно продолжать до бесконечности.

» Да, классная штука этот coreboot: быстрый, безопасный, универсальный. Можно ли его попробовать?

Можно, если у вас есть поддерживаемая аппаратура. Раньше ее список был не слишком велик, но недавно фирма AMD объявила, что все ее изделия будут выпускаться с поддержкой coreboot по умолчанию.

Чтобы узнать о coreboot побольше, выяснить, с чего начать, и просмотреть список поддерживаемой аппаратуры, посетите сайт проекта www.coreboot.org. Но помните, что замена BIOS – процесс непростой, и любая ошибка может загубить компьютер. Будьте осторожны и, прежде чем приступить, досконально изучите предмет. 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.

Во славу круглого ПО

Почему мыльные пузыри круглые? Потому что поверхностное натяжение мыльной пленки старается минимизировать площадь поверхности, а при заданном объеме наименьшую площадь поверхности имеет сфера.

Идею объема и площади поверхности можно применить и к ПО. Объемом программы можно назвать ее функциональность – то, что она умеет делать. Ну, а площадь поверхности – это то, сколько нам необходимо знать о программе, чтобы заставить ее продемонстрировать свои умения.

Современная цифровая фотокамера – пример устройства с малой площадью поверхности. С ней достаточно просто «навести и снять». (Что нужно знать, чтобы сделать фото? Сколько отдельных шагов нужно выполнить?) Тем не менее, внутри упрятаны довольно необычные и сложные штуки: автофокус, автоматическая установка выдержки и стабилизация изображения.

Думай о малом

У смартфонов площадь поверхности больше. (Что нужно знать для добавления контакта в адресную книгу? Сколько отдельных шагов нужно выполнить?) Примерно на том же уровне находятся оконные менеджеры.

На верхнем конце шкалы площадей поверхностей находятся такие вещи, как *GIMP*, *API ncurses* и редактор MIDI-файлов, с которым я сейчас бьюсь.

Понятно, что более сложные виды деятельности требуют и большей площади поверхности.

Я только хочу сказать, что большая площадь поверхности сама по себе не является преимуществом. Реакция «Ух ты! Как много настроек» может легко превратиться в «Черт! Мне что, правда нужно во все это влезать?»

При написании кода разработчики воспряют высь, и им, естественно, чужда модель мышления обычного пользователя.

А вы думайте сферически, парни. Помните мыльный пузырь.

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



Готов ко всему

BoxGrinder Создание с нуля образов виртуальных машин на базе Fedora, Red Hat и CentOS

При создании образов систем для разворачивания (скажем) на облаке EC2 вы часто начинаете с образа-шаблона, инициализируете его, затем подключаетесь к нему по SSH, чтобы добавить нужные вам пакеты.

В Fedora 15 есть утилита *BoxGrinder*, позволяющая создавать и разворачивать образы систем («устройства»), «готовые к работе» без необходимости дополнительных настроек. Она может создавать образы на основе Fedora, RHEL или CentOS для *KVM*, *Xen*, *VMware* и Amazon EC2, разворачивать их в локальной файловой системе или загружать в Amazon S3 либо как AMI (Amazon Machine Image – образ системы Amazon), либо по SFTP.

На входе *BoxGrinder* принимает простой файл «описания устройства»:

```
name: grinder-test
os:
  name: fedora
  version: 14
  password: mysecret
hardware:
  cpus: 2
  memory: 512
  partitions:
    "/":
      size: 2
packages:
  - httpd
```

В примере не показана поддержка наследования в *BoxGrinder* – можно определить «базовый» образ в файле *.appl* и затем ссылаться (и расширять) его в других файлах *.appl*.

BoxGrinder и VMBuilder

В общих чертах, *BoxGrinder* выполняет ту же работу, что и *VMBuilder* от Canonical, с той очевидной разницей, что *BoxGrinder* создает образы на базе дистрибутивов семейства Red Hat, а *VMBuilder* – на базе Ubuntu. Также *VMBuilder* не требует отдельного файла определения приложения, принимая вместо этого кучу параметров командной строки для описания системы и ее пакетов.

С этим файлом я создал образ для *VMware*:

```
# boxgrinder-build boxtest.appl -p vmware \
--platform-config type:personal \
-d local --delivery-config path:/tmp/boxtest
```

Сборка не мгновенная, так как необходимые пакеты загружаются из репозитория. В результате в каталоге */tmp/boxtest* у меня получился сжатый tar-архив с должными файлами *.vmx* и *.vmdk*, которые прекрасно загрузились на мою *VMware Workstation*. Я прервал загрузку образа на EC2, потому что на загрузку 2 ГБ по моему ADSL-каналу ушли бы не одни сутки. При создании образа системы для развертывания на облаке гораздо лучше запустить *BoxGrinder* на том компьютере в облаке, на котором вы хотите развернуть систему.

Одно из достоинств *BoxGrinder* в том, что после сборки образа для «устройства» можно перезапустить сборку для другой платформы гипервизора, и программа употребит собранный образ, не затеивая сборку заново. Подробности ищите на <http://boxgrinder.org>.

Дайте отдых компьютеру

PowerNap В Ubuntu 11.04 появилась новая версия «хранителя экрана» для сервера – рассмотрим, нельзя ли сэкономить с ней немного денег.

Когда вашему компьютеру нечего делать, сэкономить энергию можно несколькими способами: отключить ему питание или перевести его в ждущий или спящий режимы – в обоих перезапуск будет быстрее, чем запуск с нуля. А теперь есть и другая возможность – «короткий сон» [англ. пар].

Идея родилась у разработчиков Canonical Дастина Керкланда [Dustin Kirkland] и Андреса Родригеса [Andres Rodriguez], и *PowerNap* ввели в установку Ubuntu Server по умолчанию, а для других версий Ubuntu можно взять его из репозитория. Это своего рода «хранитель экрана» для серверов, но кроме активности клавиатуры и мыши он следит за «внутренней» активностью системы.

Работает он так. Демон *powernapd* через регулярные интервалы времени (по умолчанию, каждую секунду) просыпается и запускает несколько программ-мониторов, по числу типов активности системы. Если в течение заданного промежутка времени (по умолчанию – пять минут, но все эти параметры можно изменить в файле настройки) ни один монитор не фиксирует активности, демон принимает меры по снижению энергопотребления.

Мер (настраиваемых) у него четыре: отключение питания или вход в указанный режим – спящий, ждущий или энергосберегающий. Крайняя мера – отключение питания; тогда электроэнергия не расходуется вообще, но возобновление работы требует полной перезагрузки. Для этого нужно нажать кнопку включения питания или получить сетевой пакет Wake-On-LAN (WOL), если сетевой интерфейс и материнская плата поддерживают эту функцию.

В ждущем режиме компьютер также переходит в режим с почти нулевым энергопотреблением, но образ памяти сохраняется на диске, для более быстрого запуска системы. Спящий режим похож на ждущий, но память при этом не отключается, позволяя еще быстрее оживить систему. Опять же, для пробуждения компьютера нужно нажать кнопку включения питания или получить пакет WOL. Режим энергосбережения, пожалуй, наиболее интересен, хотя экономия энергии в нем самая низкая. Ради экономии применяются всякие трюки, причем компьютер готов к работе, а демон *PowerNap* продолжает следить за активностью системы – и, обнаружив таковую, работу возобновит.

Большинство низкоуровневых операций входа в ждущий, спящий или энергосберегающий режим обрабатывается набором утилит командной строки из пакета *pm-utils*. Утилиты называются по типу *pm-suspend*, *pm-hibernate* и *pm-powersave*, и действуют согласно именам. *Pm-powersave* – самая сложная из них. Она запускает скрипты из каталога `/etc/pm/power.d`, которые

- » Отключают все ядра процессора, кроме ядра 0.
- » Снижают частоту работы процессора CPU.
- » Снижают скорость работы гигабитного Ethernet до 100 Мбит/с.
- » Приостанавливают работу шины USB.

Пояснение к запросам Xpath

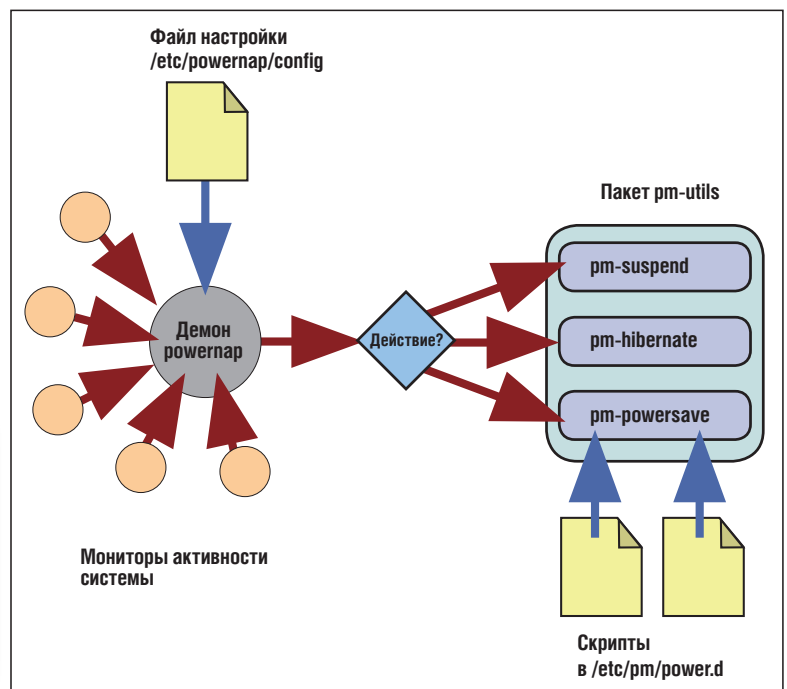
МОНИТОР	ЧТО ОН ДЕЛАЕТ
WolMonitor	Слушает сеть, проверяя наличие пакетов Wake On LAN
ConsoleMonitor	Отслеживает активность консоли (<code>tty</code>), локальной клавиатуры и мыши
ProcessMonitor	Отслеживает таблицу процессов на предмет наличия имен, соответствующих регулярным выражениям
LoadMonitor	Отслеживает пороговую нагрузку из <code>/proc/loadavg</code>
TCPMonitor	Отслеживает активность заданного набора портов TCP
UDPMonitor	Отслеживает активность заданного набора портов UDP
IOMonitor	Отслеживает активность портов ввода/вывода для заданного набора процессов
InputMonitor	Отслеживает ввод USB-устройств (клавиатура и мышь)

Заметив возобновление системной активности, демон *PowerNap* перезапускает скрипты (с аргументом `false`) для отмены «охраняемых мероприятий». Учтите: пакет *pm-utils* есть в Fedora и других дистрибутивах, а утилита *PowerNap* – пока только в Ubuntu.

Сейчас в *PowerNap* задано 8 мониторов (см. врезку вверху), хотя немалая их часть, включая монитор дисковой активности, уже на подходе. Для их включения, выключения и настройки нужно изменить параметры в файле `/etc/powernap/config`. Там также задаются интервал опроса, время бездействия, после которого система переводится в спящий режим, и другие временные параметры. Файл хорошо комментирован, и в нем есть примеры настроек, но, может быть, в пакет стоило добавить несколько образцов «профилей конфигурации» для наиболее типичных серверов?

Администраторы могут вручную инициировать и прекращать процесс «спячки» на своих серверах с помощью утилит *powernap-now* и *powerwake-now*, запуская их в локальной системе. Также есть утилита *powerwake*, используемая для отправки компьютеру WOL-пакетов через сеть.

» *PowerNap* обеспечивает логику, объединяющую усилия мониторов активности системы и утилит управления питанием.



Под итоговой чертой

В **LXF144** я обмолвился, что *PowerNap* «не потрясает», но потом решил немного посчитать «на салфетке». Дастин сообщает, что *PowerNap* сэкономил 12 Вт на четырехъядерном сервере-стойке 1u. При стоимости электроэнергии где-то в 10 пенсов/кВт*ч и линейке из 40 серверов за год вы накопили бы 420 фунтов. За эти деньги можно купить четырехгодовую подписку на *Linux Format*! Я все еще не потрясен, но впечатлен гораздо больше.

Осваиваем Grub

Grub 2 Этот загрузчик, невоспетый герой мира Linux, недавно подвергся полной переработке. Встречаем новую версию старого друга.

У каждого из нас есть программы, без которых мы как без рук – допустим, *Firefox*, *Evolution* или среда разработки *Eclipse*. Есть и другие программы, без которых нам тоже не обойтись, но мы о них особо не думаем – например, рабочий стол Gnome или ядро Linux. И остается невоспет один из настоящих героев – это *Grub*, загрузчик. Именно *Grub* отвечает за загрузку в память ядра Linux при запуске системы. Обычно он делает это быстро и без вмешательства пользователя, и легко забыть, что он вообще существует. Данный урок – моя небольшая дань *Grub*. Мы рассмотрим, как он работает и как взаимодействовать с ним, если понадобится.

Grub – сокращение от “Grand Unified Bootloader» (Великий унифицированный загрузчик), и существуют две его версии. Версия 0.97 изначально называлась просто *Grub*, но теперь ее официальное название – *Grub Legacy*. Эта версия больше не разрабатывается; релиз 0.97 был в 2005 году. Версия 1.98 (или, может быть, 1.99) официально называется *Grub 2*.

Последняя версия разрабатывается с 2007 года и была загрузчиком по умолчанию в по меньшей мере трех последних релизах Ubuntu. Но она принята не везде. В RHEL 6 по-прежнему используется *Grub Legacy* и, к моему удивлению, то же происходит в Fedora 15, хотя *Grub 2* есть в его репозиториях. Поэтому, несмотря на четырехлетний возраст, *Grub 2* все еще вызывает у некоторых ощущение «крутой новинки». *Grub 2* появился на свет в 2002 году как проект под названием PUPA (Preliminary Universal Programming Architecture [Предварительная универсальная программируемая архитектура] – расшифровка из старомодных, если я вообще их слышал!) Меня так и тянет сказать, что если *Grub Legacy* был PUPA [англ. куколка (насекомого)], он должен бы превратиться в MOTN [англ. мотылек, бабочка], а не в *Grub 2*.

Картина в целом

Первая программа, которую компьютер запускает при загрузке – это BIOS, она хранится в ПЗУ на материнской плате. Если компьютер загружается с жесткого диска, BIOS загружает главную загрузочную запись (MBR – Master Boot Record) – первый сектор жесткого диска. Небольшая программа, размещенная в MBR загрузчиком *Grub*, используется для следующего этапа загрузки – загрузки модуля *Grub core.img*.

В MBR недостаточно места для размещения программы, способной интерпретировать файловую систему Linux, поэтому туда встроены номера физических блоков, где расположен файл *core.img*.

Этот файл – действительно сердце *Grub*. Его задача – найти образ ядра Linux и образ *initrd* в файловой системе на жестком диске, загрузить их в память и передать управление ядру. Обычно он считывает файл настройки (*/boot/grub/grub.cfg*), где определено число записей в загрузочном меню. (Даже если задано всего одно загружаемое ядро, будет показан второй вариант для его загрузки в «безопасном» режиме).

Обычно взаимодействие с *Grub* – если оно вообще происходит – осуществляется через интерфейс меню. Но у *Grub* есть похожий на оболочку командный интерпретатор, о котором я сперва и расскажу. Во время загрузки можно зайти в командную строку *Grub* из меню загрузки, нажав клавишу С. (В зависимости от конфигурации GRUB может потребоваться в нужный момент нажать Esc, чтобы увидеть меню загрузки). *Grub* также входит в ре-



► *Grub* позволяет изменить команды, заданные в *grub.cfg*, до их вызова для загрузки системы.

жим командной строки, если не найден файл настройки. (Прежде чем вы в поисках приключений рванулись удалять на пробу свой *grub.cfg*, предупреждаю, что вероятнее всего вам в итоге придется прибегнуть к загрузочному диску, чтобы вернуть этот файл обратно).

В своей командной строке с подсказкой **grub>** интерпретатор предлагает более 150 встроенных команд. Команды

```
grub> set pager=1
grub> help
```

позкранно выведут их список. Чтобы получить подробную справку (например) про команду *search*, попробуйте команду:

```
grub> help search
```

С помощью всего пяти из этих встроенных команд можно вручную указать для загрузки образ ядра и соответствующий *initrd* следующим образом:

```
1. grub> insmod ext2
2. grub> set root="(hd0,1)"
3. grub> linux /boot/vmlinuz-2.6.38-8-generic root=/dev/sda1 ro
4. grub> initrd /boot/initrd.img-2.6.38-8-generic
5. grub> boot
```

Рассмотрим эти команды по очереди. Строка 1 загружает модуль (*/boot/grub/ext2.mod*), который понимает файловые системы *ext2/3/4*. Это всего один из примерно 160 динамически загружаемых модулей *Grub*. В строке 2 задается корневой раздел *Grub* (*root*). Как видите, *Grub* применяет собственные обозна-

Начальный RAM-диск

Grub также отвечает за загрузку образа диска начальной инициализации, файла, где хранится небольшая файловая система ОЗУ. Она содержит загружаемые модули ядра и другие компоненты, необходимые ядру на ранних этапах процесса загрузки – до того момента, когда возможно монтирование раздела *root*. Файл представляет собой сжатый архив *cpio*. Для его просмотра используйте следующие команды, соответственно заменив во второй строке имя файла:

```
cd /tmp
cp /boot/initramfs.img junk.gz
gunzip junk.gz
cpio -it < junk.gz
```

Вы удивитесь, как много здесь всего.

Grub Legacy и Grub 2

- » Файл настройки (**grub.cfg**) сильно отличается от старого **menu.lst**, и его синтаксис гораздо ближе к полноценному скриптовому языку.
- » Для генерации **grub.cfg** есть утилиты; обычно редактировать его вручную не приходится.
- » **Grub 2** умеет идентифицировать разделы по меткам файловой системы или UUID.
- » **Grub 2** умеет читать файлы прямо с устройств LVM и RAID.
- » Файлы образов полностью изменены. Старые этап 1, этап 1.5 и этап 2 исчезли. Многие возможности теперь реализованы в динамически загружаемых модулях.
- » Доступны графический терминал и графическая система меню с обширной поддержкой тем оформления.
- » **Grub 2** поддерживает и другие архитектуры загрузки, кроме BIOS ПК, включая PC EFI, Power PC и SPARC.

чения для именованного раздела – (**hd0,1**), вероятно, соответствовало бы **/dev/sda1** в терминах Linux. (Если вы привыкли к **Grub Legacy**, обратите внимание, что разделы теперь нумеруются с 1, а не с 0.) Строка 3 загружает образ ядра Linux. Имя файла интерпретируется по отношению к разделу **root**, заданному в предыдущей строке, и если вы печатаете с ошибками, как я, то будете рады узнать, что в **Grub** реализовано автодополнение имен файлов в стиле **Bash**. Аргументы, следующие за именем файла, передаются ядру как «аргументы командной строки». Строка 4 загружает образ «диска начальной инициализации» [initial ram disk], и, наконец, в строке 5 управление передается ядру Linux. Работа **Grub** завершена.

Разумеется, все эти команды обычно не приходится набирать при каждой загрузке Linux. Обычно **Grub** считывает свой файл, выводит меню загрузки на основе того, что он нашел, и выполняет команды согласно выбранному пункту меню в файле настройки. Вот строки из файла **grub.cfg**, соответствующие введенным ранее вручную командам:

```
menuentry 'Simplified Boot' {
    insmod ext2
    set root='(hd0,1)'
    linux /boot/vmlinuz-2.6.38-8-generic root=/dev/sda1 ro
    initrd /boot/initrd.img-2.6.38-8-generic
}
```

На практике записи в **grub.cfg** в Ubuntu выглядят немного сложнее. Основное различие в том, что Ubuntu предпочитает идентифицировать свои разделы по UUID, а не по имени устройства Linux. Поэтому вы увидите добавочные команды **Grub**, задающие корневой раздел по UUID. Но мой упрощенный пример превосходно работает.

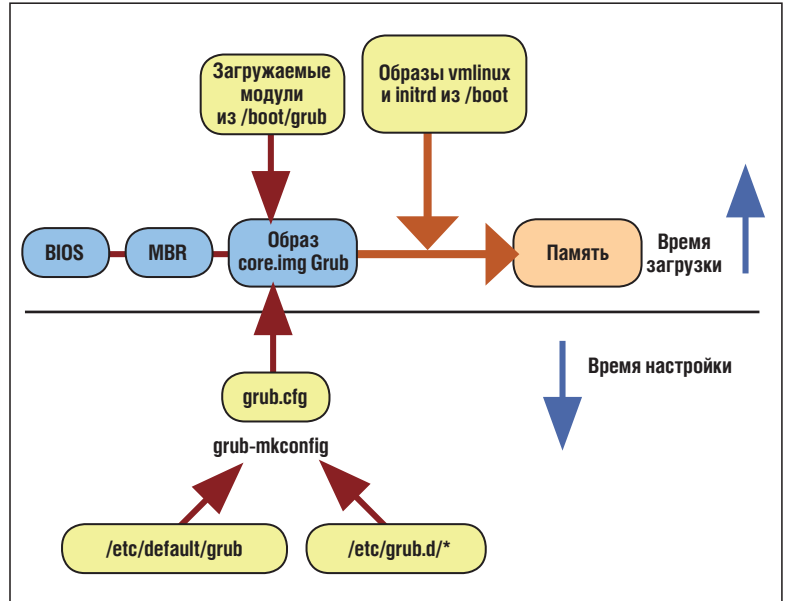
Синтаксис файла настройки по сути представляет собой полноценный скриптовый язык, поддерживающий переменные, циклы и условия, во многом в духе **Bash**.

Создание файла настройки

Я проверил приведенный выше пример, редактируя файл **grub.cfg** вручную. Но обычно стоит для этого запустить скрипт **grub-mkconfig**. Он проверит вашу систему на предмет загружаемых образов и сгенерирует пункты меню для создания рабочего файла **grub.cfg**, и даже найдет засидевшуюся у вас установку Windows и добавит пункт меню для нее.

Подробности работы данной схемы немного сложны. Во-первых, в файле **/etc/default/grub** определяется набор параметров, управляющих работой **grub-mkconfig** (см. врезку).

Кроме того, **grub-mkconfig** запускает скрипты из каталога **/etc/grub.d**. Вывод этих скриптов включается в созданный **grub.cfg**. Например, скрипт **30_os_prober** обследует разделы на наличие загружаемых образов ядра и добавляет в **grub.cfg** записи для



их загрузки. В **/etc/grub.d** можно добавить собственные скрипты, если вы хотите разместить свои пункты меню в **grub.cfg**.

Grub может напрямую загружать множество операционных систем, включая Linux и различные версии BSD. Для других систем, типа Windows, **Grub** использует «загрузку по цепочке», т. е. вызывает «родной» загрузчик Windows с соответствующего раздела. Вот запись для этого пункта меню в моей системе с двойной загрузкой:

```
menuentry "Windows 7 on /dev/sda3" {
    insmod ntfs
    set root='(hd0,3)'
    chainloader +1
}
```

Таинственный аргумент **+1** – несколько вырожденный пример списка блоков. Он означает «загрузить 1 блок, начиная с блока 0 на корневом разделе» (в этом примере – **/dev/sda3**).

Существует компромисс между вводом собственных команд **Grub** вручную и выбором пункта меню. В меню загрузки выберите пункт и наберите **e**. Вы попадете в редактор в стиле **emacs** и сможете изменить команды из файла настройки. Он часто используется, например, при добавлении параметра **single**, для загрузки в однопользовательском режиме.

Grub 2 – довольно сложная зверюга. Чтобы узнать о нем больше, скомандуйте **info grub** или загрузите руководство с сайта <http://www.gnu.org/software/grub/manual>. **LXF**

» **Grub** отвечает за вызов ядра Linux в память во время загрузки системы, под управлением автоматически сгенерированного файла настройки.

Пояснение к параметрам

GRUB_DEFAULT	Пункт меню по умолчанию (он будет выбран, если пользователь ничего не выберет за отпущенный период времени).
GRUB_TIMEOUT	Промежуток времени в секундах, в течение которого будет отображаться меню.
GRUB_SAVEDEFAULT	Если "true", то выбранный в этот раз пункт меню станет пунктом меню по умолчанию при следующей загрузке.
GRUB_DISABLE_LINUX_UUID	Если "true", grub-mkconfig перестает определять корневой раздел в Linux по UUID. Вместо этого используется простое имя устройства Linux.
GRUB-INSTALL	Устанавливает Grub на устройство.
GRUB-MKIMAGE	Создает core.img , содержащий заданный список модулей.
GRUB-MKRESCUE	Создает ISO-образ загрузочного аварийного CD Grub .
GRUB-MENULST2CFG	Конвертирует файл menu.lst Legacy в grub.cfg .
GRUB-MKNETDIR	Копирует дерево каталогов Grub на TFTP-сервер, чтобы Grub можно было бы загружать через сеть (с использованием загрузки PXE).



КНИЖНАЯ ПРЕМИЯ РУНЕТА 2011

■ Художественная литература ■ Non-fiction ■ Бизнес-книга ■ Детская литература

WWW.OZON.RU/PREMIA

Голосуй, если ты читаешь!

1.07.2011 - 20.08.2011

Награждение лауреатов на ММКВЯ-2011

8.09.2011

Контакты: www.ozon.ru, premia@ozon.ru

OZON.RU



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул PostgreSQL на защите кандидатской диссертации по физике элементарных частиц.

Свободное аппаратное обеспечение

Чужими руками хорошей работы не сделаешь.
Петр Леонидович Капица

Идея открытого аппаратного обеспечения родилась вместе с электроникой. Точнее, даже и идеи не было: просто вместе с телевизором поставлялась его принципиальная схема. Когда же бытовая электроника окончательно стала массовым продуктом, принципиальную схему стали зажимать. Права потребителя незаметно для большинства деградировали.

Причина такого процесса в том, что данное большинство этими правами не пользовалось. Поэтому теперь идею приходится возрождать фактически с нуля. Правда, время не прошло даром, и от идеи открытого аппаратного обеспечения отпочковалась идея свободного железа. Вы не просто имеете право посмотреть документацию – вы имеете право собрать железку по ней.

CERN (это там, где сейчас эксплуатируют Большой адронный коллайдер и где придумали WWW) предложил сообществу CERN Open Hardware License версии 1.1 вместе с хранилищем проектов (<http://www.ohw.org/>), которые распространяются под этой лицензией. CERN – большая организация, и БАК, а также другие эксперименты, будут работать долго, поэтому CERN кровно заинтересован в свободном доступе ко всем типовым схемам, используемых для управления БАКом и сбора с него данных. Если что-то с подрядчиком случится, то в случае лицензии CERN OHL всегда можно найти другого. Это получается совершенно естественно.
E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Сажать компьютер на диету 56
Тем, на чей компьютер взвалена непосильная тяжесть, **Боб Мосс** подгонит LXDE и Fluxbox, двух чемпионов в весе пера.



Править фото 60
Следуйте за **Адамом Оксфордом** – и вы приведете свои снимки в более чем божеский вид: в бесплатной фотостудии Linux есть для этого все.



Рулить процессами 64
В предпоследнем выпуске серии **Майк Сондерс** покажет, как приструнить отбившиеся от рук процессы и стать виртуозом Vi.



Делиться файлами 68
Хранить данные в Интернете надежно, выгодно и поразительно удобно. **Эндрю Грегори** демонстрирует, на что тут способен Linux.



Следить за сайтами 72
Учитесь у **Тихона Тарнавского** узнавать об изменениях в Сети, не прикладывая браузера – да, командная строка умеет и такое!



Вникать в Erlang 76
И в функциональном языке программирования есть что коллекционировать. **Андрей Ушаков** начинает с коллекций-списков.



Делать C++ читаемым 80
В этом занятии вам непременно помогут практические и полезные библиотеки из набора Boost и **Семен Есилевский**.



Настраивать Gnome Shell 84
Установки по умолчанию вам не нравятся? Расстраиваться незачем – с **Джонатаном Робертсом** вы мигом все преобразите.

Совет месяца: Узнаем IP-адрес



Если в вашей сети, как у большинства, используется DHCP, узнать IP-адреса, отведенные устройствам, не так-то просто.

Можно бы скомандовать в терминале `ifconfig` – ну, а если у вас свежеподключенный сервер без монитора и клавиатуры? Тогда без IP-адреса на него по SSH не выйти.

А вы загляните в файлы журнала маршрутизатора – в предположении, что они есть и вам разрешено это делать.

Одно из решений – таблица `arp`, применяемая для установки соответствия MAC- и IP-адресов, а MAC-адрес на устройствах обычно нанесен.

Команда `sudo arp -n` покажет текущую таблицу ARP, и если устройство уже «в контакте»,

вы все там увидите. Если нет, принудительно задайте ему временный IP-адрес – это позволит вам войти через SSH и задать конфигурацию сети для конкретного хоста при установке соединения с вашего компьютера, при помощи команды

```
arp -s IPaddress MACaddress
```

Если вы – обладатель прав на сканирование сети, вам очень пригодится утилита `netdiscover` (<http://nixgeneration.com/~jaime/netdiscover>), запускаемая командой

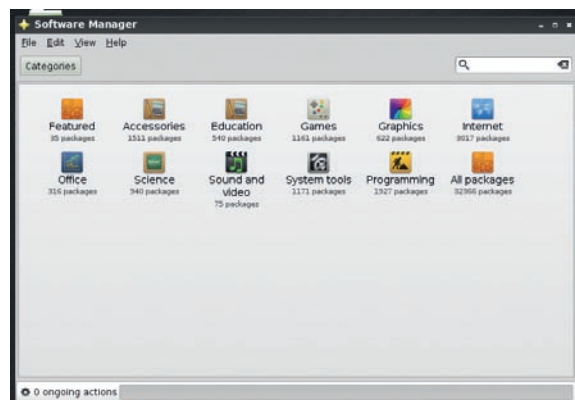
```
sudo netdiscover -i eth0
```

Вы существенно сократите время ожидания результата, указав утилите диапазон просмотра – параметром -r:

```
sudo netdiscover -i eth0 -r 192.168.1.0/24
```

Mint LXDE: Легко

Ищете облегченный вариант ОС, чтобы вернуть к новой жизни старый компьютер? **Боб Мосс** представляет обзор трех дистрибутивов-легковесов.



► Расширьте диапазон своего программного обеспечения с помощью исключительно полезного менеджера пакетов.

Linux – этот дистрибутив пользуется заслуженной популярностью на форумах нашего сайта и блестяще отражает среду *Fluxbox* рабочего стола.

А для тех, кому нужен дистрибутив, в хорошем смысле анорексичный, мы бегло оглядим Damn Small Linux – систему, которая ухитряется работать на машинах с жалкими 8 МБ оперативной памяти (каких не бывало уже со времен Windows 95) – хотя, если вы предпочитаете Linux-установку с более новыми инструментами и приложениями, вероятно, разумно будет припасти по крайней мере 16–32 МБ ОЗУ.

В зеркалах

Версии упомянутых дистрибутивов включены в диск этого месяца, но при желании найти последний и самый замечательный ISO-образ для Mint LXDE зайдите на <http://www.linuxmint.com/edition.php?id=87>.

По стандартам Ubuntu, шагов установки на удивление мало, и даже на очень скромном по современным меркам оборудовании все завершится гораздо быстрее, чем стандартная установка Gnome/KDE.

Установив его и подготовив к работе, приступим к изучению. Внизу слева вы найдете LXDE Launcher с четкой категоризацией приложений, предоставляемых по умолчанию. Наиболее очевидная категория – Интернет, откуда запускаются *Firefox*, *Gwibber* и торрент-клиент *Transmission* (см. учебник прошлого месяца). Вы также найдете текстовый процессор *Abiword* и электронные таблицы *Gnumeric* – см. врезку «Живите свободно», чтобы узнать, как установить более громоздкий *LibreOffice*.

Есть ряд отличных инструментов, которых новые пользователи часто не замечают. Первый – это профилирование железа, помещенное в разделе меню System Tools. Выходцам с Windows



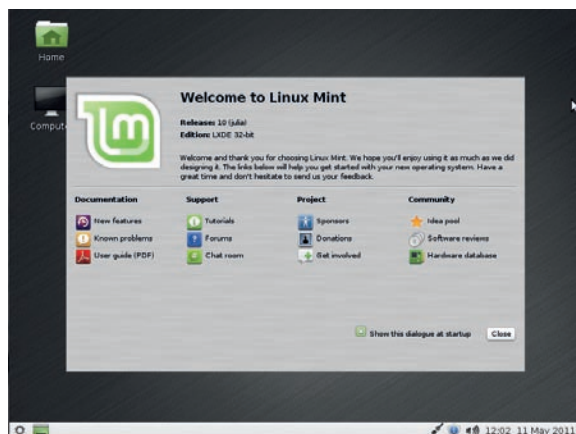
Наш эксперт

Боб Мосс разрывается между написанием кода и учебой по курсу компьютерных наук.

В прошлом месяце мы рассмотрели *Xfce* – легкий и быстрый рабочий стол, способный работать на гораздо более древнем аппаратном обеспечении, чем Gnome 3 и KDE4 – рабочие столы основных Linux-дистрибутивов.

На сей раз мы рассмотрим два рабочих стола – *LXDE* и *Fluxbox*, на фоне которых даже *Xfce* выглядит как подсевший на курение ветеран сумо посреди марафонской дистанции.

В первой части нашего урока мы рассмотрим *LXDE* в дистрибутиве Mint LXDE от Клема Лефевра [Clement Lefebvre]. Вторую половину урока мы посвятим более близкому знакомству с Puppy



► Если, по-вашему, Unity, Gnome и KDE слишком увесисты, вернитесь к основам с *LXDE*.

► **Месяц назад** Мы освоили *Xfce* и настроили его на свой вкус.

работается



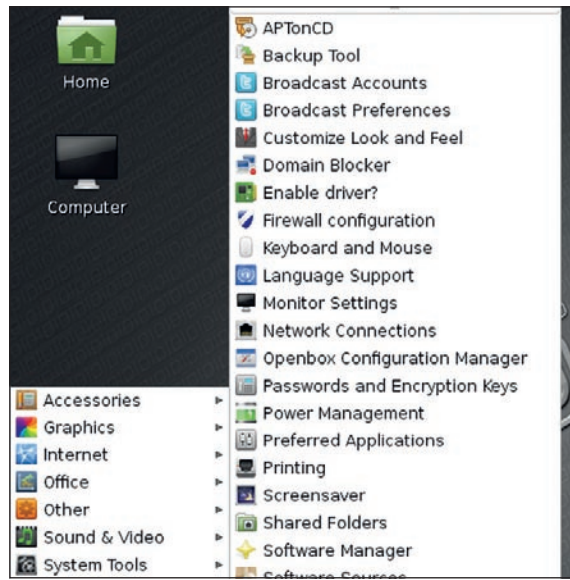
оно покажется до боли знакомым, и здесь вы обретете всю полноту информации о вашей системе, от разрешения экрана и загружаемых модулей ядра до активных IP-соединений.

Самая интересная часть находится под Benchmarks на левой панели: в ней вы можете оценить производительность своего процессора при помощи целого ряда тестов, а затем посмотреть прямое сравнение с другим процессором того же поколения.

В этом же разделе меню Launcher помещен также Диспетчер задач [Task manager]. Здесь вы узнаете о текущем потреблении CPU и оперативной памяти, а также пресечете нежелательные процессы, выбрав их и нажав Удалить.

И, наконец, для изменения внешнего вида вашей системы достаточно щелкнуть правой кнопкой мыши на рабочем столе и выбрать Настройки рабочего стола. Здесь вы можете изменить обои, а также шрифт и цвет системных сообщений.

Для более тонкой настройки перейдите в раздел Настройки системы через меню Launcher и выберите пункт Настроить внешний вид. В нем настраиваются наборы иконок, рабочий стол, курсор мыши и множество других элементов внутри вашей установки LXDE.



» LXDE хоть и выглядит пустынным, но прекрасно настраивается, а Mint предлагает отличное меню Launcher.

Скорая помощь

В Mint LXDE предусмотрена установка несвободных драйверов, через апплет Launcher > Настройки > Установить драйвер?.

Пингвины в кармане

Возможно, вас шокировало, что священные страницы *Linux Format* пятнает экранный снимок Windows 7, однако на то есть веские причины. Легковесные дистрибутивы, которые мы рассматриваем в этом уроке, требуют всего 32 МБ (хотя в случае Mint LXDE вам скорее потребуется минимум 2–3 ГБ).

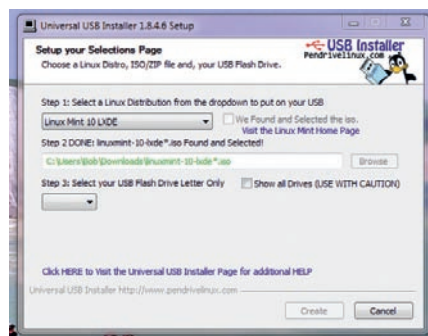
Добавьте еще незначительный объем ресурсов, потребляемых этими дистрибутивами, и превосходную поддержку железа в Linux – и вот вам потенциал для легковесного «портативного» дистрибутива Linux, уместяющегося в кармане.

Но что мы понимаем под портативностью?

Старый способ добиться портативности – носить с собой Live CD для загрузки системы с него, но этот способ не лишен минусов: диск легко поцарапать, и на CD нельзя добавлять пакеты или сохранять файлы.

Теперь у нас есть гораздо более эффективная альтернатива – установка Linux-дистрибутива на USB-брелок. Этот не сломается и не поцарапается (если не наступать на него ногой), и позволяет сохранять документы и обновленную копию дистрибутива.

Популярные дистрибутивы вроде Ubuntu и Fedora уже предлагают встроенный инструмент для создания загрузочных USB-версий самих себя (как правило, в системе меню рабочего стола Gnome). Менее известные дистрибутивы иногда также пре-



» Перейдите на www.pendrivelinux.com, чтобы легко и быстро установить дистрибутив Linux на USB-накопитель.

доставляют подобное, но их продукты подвержены тенденции проб и ошибок.

В Puppy Linux утилита находится в главном меню запуска под Настройка > Установить Puppy на USB. При установке просто следуйте инструкциям мастера.

В Damn Small Linux также есть способ сделать это с помощью загрузчика – перейдите в Приложения > Инструменты > Установить USB-HDD Pendrive; но пользователям, незнакомым с разбивкой на разделы и взаимодействием Linux с различными

устройствами, или предпочитающим не соваться в терминал, мы рекомендуем держаться от этого инструмента подальше.

Linux Mint LXDE также предлагает собственный мастер, но он, похоже, требует от пользователя скачать ISO-образ Mint LXDE еще раз (если только у вас нет Live CD).

Так почему же на рисунке – Windows 7? Ну, самые надежные инструменты для создания загрузочных Linux-флэшек, как правило, приходят из www.pendrivelinux.com, а они склонны работать в Windows.

Но прежде чем в омерзении выплевывать свое горячее питье, вы сообразите, что если вы сейчас пользователь Windows, или уже работаете с двойной загрузкой, или вам просто нужна быстрозагружаемая система для починки сломанного компьютера, то у вас, вероятно, эти системы еще не установлены (а стало быть, наличие собственных инструментов для таких целей роли не играет).

Как видно по рисунку, вы просто указываете, какой дистрибутив установить, и Мастер, как правило, находит ISO-образ в папке Downloads.

Выберите букву диска из выпадающего меню, нажмите кнопку Далее и следуйте дальнейшим инструкциям. Через несколько минут вы должны получить очень портативную и легкую загружаемую среду Linux.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Puppy Linux: Еще

Если вы решили еще отощать, попробуйте совсем уж диетические варианты.

Если Mint LXDE для вас недостаточно мал, можете выбрать еще менее тяжелый, взглянув на Puppy Linux. В качестве среды рабочего стола этот дистрибутив использует почтенный *Fluxbox*, и разработчики подошли к составу пакетов очень серьезно.

По умолчанию в системе установлен единственный браузер – *Dillo* (см. фото в «Шаг за шагом», на смежной странице), но по двойному щелчку на значке Обзор на рабочем столе вам предложат браузеры на выбор.

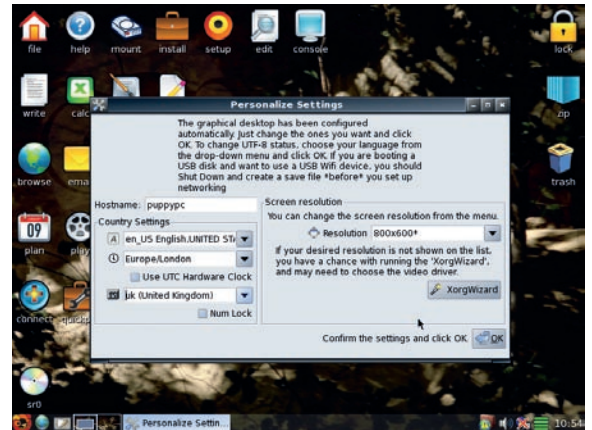
Три браузера вам, вероятно, знакомы – *Firefox*, *Opera* и *Chrome*; но есть также *Seamonkey* (ныне почти вымерший) или *Iron* (производное от *Chromium*). Просто выберите, какой вам больше нравится, и он станет браузером по умолчанию.

Для обработки документов есть обязательные *Gnumeric* и *Abiword* (см. врезку «Живите свободно», если вы предпочитаете *LibreOffice*).

Самое интересное – впереди. Например, вы можете подумать, что показанный на рисунке менеджер пакетов – единственный способ установки ПО; однако, нажав на значок Quickpet на рабочем столе, вы найдете меньшую, но гораздо более удобную альтернативу. Отсюда можно скачать несвободные драйверы и пакеты, а также часто используемые приложения вроде *Inkscape* и *VLC*.

Кроме того, если у вас возникли проблемы с подключением к Интернету, щелкните на значок Подключения – и появится удобное всплывающее окно с ассортиментом разнообразных мастеров.

Для настройки Wi-Fi соединения нажмите на значок рядом с Интернет Через Проводную или Беспроводную Локальную сеть в разделе Подключение, а затем нажмите Простой мастер сети, выберите свой Wi-Fi-интерфейс и заполните данные сети (если это вызовет осложнения, попробуйте использовать мастер сети по умолчанию – в нем больше опций настройки). По той же схеме



» Зачем затягивать нудную процедуру установки, когда все важные настройки можно сделать при первом запуске?

проводится подключение через Ethernet (в большинстве случаев в качестве сетевого интерфейса сгодится eth0).

И, наконец, чтобы изменить внешний вид своего рабочего стола, щелкните правой кнопкой мыши на рабочем столе, затем поведите курсором у Установки рабочего стола. Здесь огромное количество инструментов; некоторые из них имеют странные неоднозначные имена.

Полезнее всех из них *Nathan Wallpaper Setter*, позволяющий менять обои для рабочего стола; *ChTheme GTK Theme Chooser*, изменяющий вид *GTK-приложений* во *Fluxbox*; *Configure XLOCK Screen Locking*, где можно определить предпочтительный хранитель экрана; *Desktop Icon Switcher*, выполняющий импорт наборов иконок с других настольных ПК, и *Set Global Font Size*, задающий размер системного шрифта.

Скорая помощь

Приложения запускаются во *Fluxbox* с помощью Launcher вверху слева или щелчком правой клавишей на любом месте рабочего стола.

Перейдем к облакам

Настанет момент, когда вы захотите использовать web-почту и онлайн-хранилища наподобие *Dropbox* или проверить свои учетные записи на Facebook и Twitter. И Mint LXDE, и Puppy Linux в полной мере к этому готовы.

В случае Mint LXDE в качестве почтового клиента предустановлен и готов к настройке *Thunderbird*. Puppy Linux использует легкий почтовый клиент под названием *Sylpheed*. К сожалению, у нас здесь мало места для подробных инструкций (хотя мы рассказывали об установке почтового клиента в **LXF145**, доступном для подписчиков через наш сайт), но если вы создаете учетную запись Gmail, потребуются следующие настройки:

- » Настройки, требуемые Gmail
- » Настройки IMAP
- » Сервер входящей почты: **imap.gmail.com**
- » Порт сервера входящей почты: **993**
- » Шифрование: **SSL**
- » SMTP Settings
- » Сервер исходящей почты: **smtp.gmail.com**
- » Порт сервера исходящей почты: **465**
- » Шифрование: **TLS**

В случае *Dropbox* вы обнаружите, что Mint LXDE настроить гораздо легче, так как он обрабатывает Deb-файлы, и можно скачать *Mint Dropbox* прямо с сайта *Dropbox*. Однако на момент написания нельзя нажать правой кнопкой мыши в файловом менед-

жере для копирования общей ссылки, хотя вполне возможно, что это станет доступно ко времени чтения вами данной статьи. Если вы хотите установить *Dropbox* на Puppy Linux, будьте готовы к довольно жаркой поре; установка *binary-only*, похоже, работает не на всех моделях, и по-любому требует значительного числа зависимостей. При желании поднажать полезной может быть эта ветка форума: <http://bit.ly/k56Rns>.

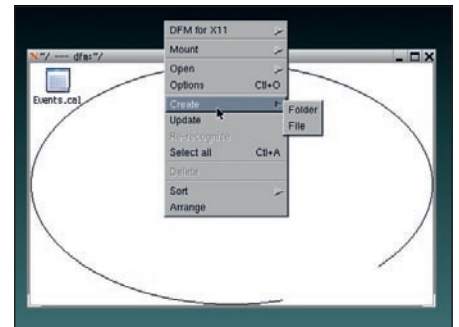
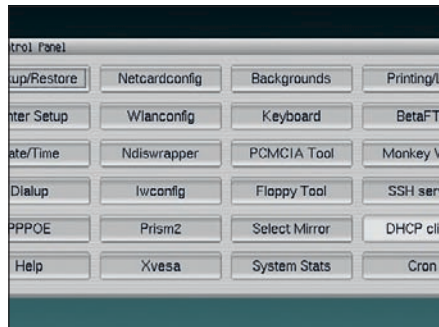
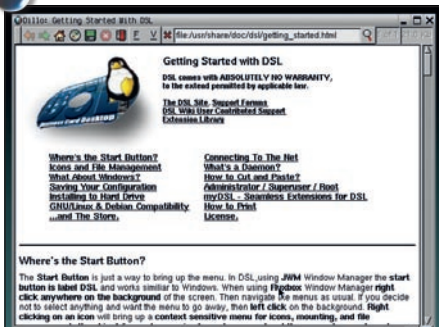
Наконец, вы обнаружите, что Mint LXDE включает *Gwibber* для слежения за оповещениями Facebook и Twitter. Puppy Linux еще не обеспечил это по умолчанию, но *Gwibber* доступен через менеджер пакетов.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

ТОГО ЛЕГЧЕ...



Шаг за шагом: Damn Small Linux (DSL)



1 Читаем документацию

Сразу же при запуске появится *Dillo* и предоставит вам документацию о наиболее частых вопросах пользователей по DSL.

2 Настроим систему

Любые настройки или конфигурации, которые вам, вероятно, потребуется изменить, находятся на особой панели управления DSL.

3 Управляем файлами

Минимализм файлового менеджера не означает, что в программе не хватает функций. С DSL вы все равно можете выполнить почти все ваши ежедневные операции.



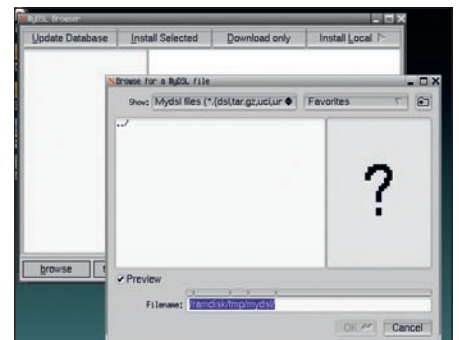
4 Просмотр сайтов

Да, это тот браузер, про который вы подумали. Урезанная версия *Firefox* доступна изначально, но для дополнений потребуется добавочная память.



5 Меню навигации

В меню DSL – множество ПО и апплетов панели управления. *Siag* обеспечивает удивительно богатую функциональность электронных таблиц.



6 Установка пакетов

Если вы хотите расширить свою установку, MyDSL предоставляет интерфейс менеджера пакетов на базе web. Можно также установить *MyDSL* и пакеты исходников. [LXP](#)

Живите свободно

Хотя *Abiword* и *Gnumeric* обеспечивают поразительную функциональность «малой кровью», вы можете возжелать набор функций пошире и совместимость документов, предоставляемые новым *LibreOffice*.

Пользователи последних версий основных дистрибутивов уже встречали этот пакет в том или ином облике, но если вы еще не сталкивались с ним – это оптимизированная версия почтенного *OpenOffice.org* с добавочными функциями, доступными под «более свободным» лицензионным соглашением.

Поскольку Oracle в настоящее время приостановил проект *OpenOffice.org*, *LibreOffice*, безусловно, стал ориентиром для офисных пакетов, и он может работать на вашей новой сверхбыстрой «обезжиренной» системе молниеносно.

Во время написания статьи *LibreOffice* отсутствовал в репозиториях Mint LXDE. Для его установки зайдите на <http://www.libreoffice.org/download> и выберите установку Debian. Позаботьтесь установить пакет Интеграции с рабочим столом (*GTK*),

чтобы пакет появился в вашем меню *LXDE* (заметим, что на момент написания поддержка была несовершенна, и может оказаться, что некоторые элементы *LibreOffice* выглядят менее приятно, чем должны бы).

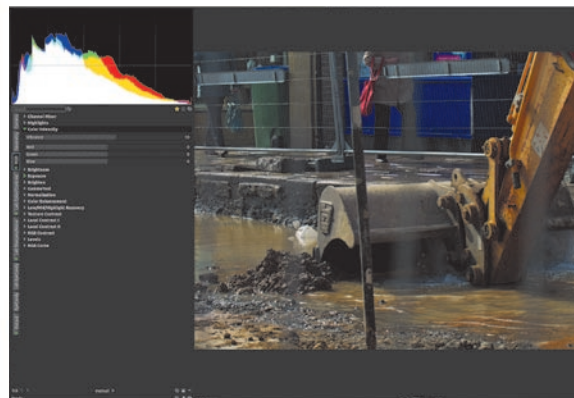
Для Puppy Linux необходимо обычным порядком скачать и установить исходный код (см. README внутри архива для более подробной информации). Кроме того, доступна упакованная версия на <http://www.puppylinux.org/wikka/LibreOffice>.

» Через месяц Разберемся, как запускать приложения Windows в Linux.



Linux для

Адам Оксфорд опровергает устоявшееся мнение, доказав, что даже серьезно занимаясь фотографией, не стоит пренебрегать любимой ОС.



➤ **Photivo** — многообещающее приложение для обработки RAW-файлов. Оно изумительно мощное, но устрашает сложностью.

(Blue) каналы на пиксель, или по 256 уровней яркости на канал. Это знакомый набор 16,77 млн возможных цветов, где черный выражается как значение RGB (0,0,0), а белый – (255,255,255). А вот RAW-изображения содержат 12 или 14 бит на канал, или до четырех триллионов цветов. Разница между одним и тем же изображением при 8-битной и 14-битной глубине цвета невооруженным глазом не воспринимается – и не отображается на мониторе компьютера.

Необработанный талант

Фотографам важно то, что при простом редактировании – типа настройки уровней выдержки при помощи гистограммы в *GIMP* – 8-битное изображение скоро выдаст артефакты или цветные полосы, а с 12-или 14-битным RAW-файлом или 16-битным TIFF вы можете измываться над фотографиями, презрев Женевскую конвенцию, и все же получать сглаженные, плавные градиенты цветов и четкие края – или почти что так.

Все зеркальные фотокамеры и многие компактные камеры могут снимать в формате RAW. Каждый изготовитель применяет собственный тип файлов для сохранения этого формата. Изначально это делалось для того,

чтобы привязать фотографов к своему ПО для конвертации; а теперь, вероятно, просто из вредности. Существует масса программ, способных читать RAW-файлы со всех

камер, большинство из которых – в том числе стандарт индустрии *Adobe Camera RAW* – основаны на открытом исходном приложении *Dcraw* (<http://www.cybercom.net/~dcoffin/dcraw/>); оно позволяет читать практически любые типы файлов RAW, благодаря способностям ее создателя в обратной инженерии.

Dcraw – разработка Дэйва Коффина [Dave Coffin], поддерживаемая им добрых десять лет; возможно, это один из наиболее весомых актов альтруизма в мире фотографии.

Большинство возражений по поводу Linux основаны на слепом предубеждении или неосведомленном убеждении, что ничто не изменилось в мире KDE или Gnome с прошлого века. И даже самые пылкие пингвинофилы не горят идеей применить Ubuntu или Fedora для серьезных занятий фотографией.

Но! Если вы знаете, где искать инструменты и как их использовать, вы будете работать так же эффективно и получать результаты не хуже, чем в известнейших редакторах фото, ничуть не поступаясь принципами философии свободного и открытого исходного кода.

Первое, что следует сделать – прекратить использование JPG-файлов: фото надо брать в формате RAW [англ. необработанный, сырой]. Это «цифровой негатив» – несжатые данные, захваченные непосредственно с датчика камеры. RAW-файл намного увесистее, чем JPG, произведенный встроенным в камеру процессором обработки, но если вы никогда не работали с RAW, вы поразитесь, как много теряется от преобразования фотографий в JPG-файлы.

Даже на дорогих цифровых зеркальных фотокамерах динамический диапазон JPG на выходе ниже, чем у RAW-файлов, а значит, пропадают светлые и темные детали, которые можно раскрутить в итоговом изображении.

При конвертации в JPG со сжатием некоторые детали утрачиваются из-за возникающих артефактов и усиления резкости, но самая большая разница с технической точки зрения – в цветовой гамме. В JPG-файлах используется классический стандарт экранного формата: по 8 бит на красный (Red), зеленый (Green) и синий

«Первое, что следует сделать — сменить формат JPG на RAW.»



Наш эксперт

Адам Оксфорд

Бывший редактор *Digital Camera Shopper* и большой фанат Linux и всего, что касается открытого исходного кода.

фотографов

Благодаря вездесущести *Dcraw*, есть много инструментов FOSS, умеющих производить изображения из RAW-файлов фотографий такого же качества, что и те, которые выходят из какого-нибудь *Adobe Lightroom* или *Aperture* от Apple.

Одно из многих ключевых преимуществ съемки RAW – то, что программы, «проявляющие» RAW-изображения и конвертирующие их в файлы TIFF или JPG для дальнейшей обработки, не разрушают исходное изображение. Все примененные изменения сохраняются в виде текстового файла-инструкции, выполняемой только при сохранении картинки в формате JPG или TIFF.

Если вы намерены снимать сотни фотографий, у вас не будет времени на открытие каждой в программе для обработки изображений, поэтому вам нужно наладить рабочий процесс.

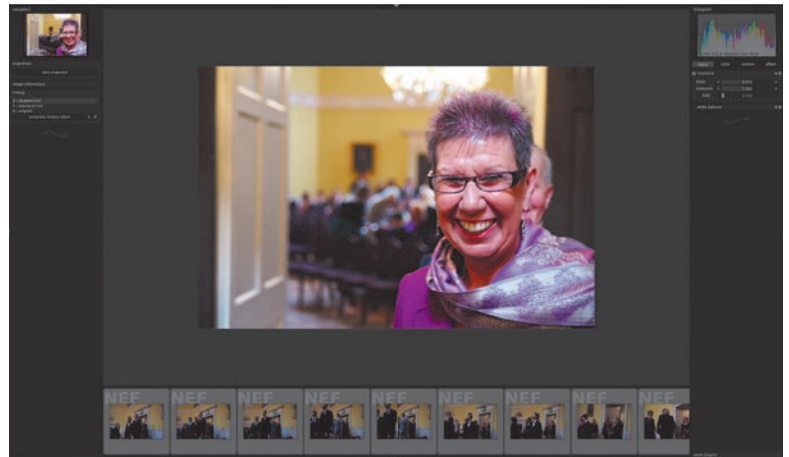
В фотографии рабочий процесс делится на четыре этапа. Первый – это отбор фотографий из всего отснятого материала. Затем нужно «проявить» изображения, используя программное обеспечение для преобразования. Следующий шаг – вывод изображений и тонкая их настройка в графической программе манипуляции, типа *GIMP* или *Photoshop*. Наконец, фото нужно подготовить к печати, отправке в Сеть или записи на DVD, в зависимости от того, что вы собираетесь с ними делать в дальнейшем.

Некоторые профи предпочитают специализированные программы для каждого шага, а те, кто хочет добавить весьма специфические эффекты, способны использовать для каждого шага два или три различных приложения.

Импорт изображений, например, можно осуществить всего лишь простым их перетаскиванием с кард-ридера в файл на жестком диске. Вам потребуется определенный метод именования большого количества снимков – это можно осуществить с помощью простого скрипта (**LXF143**) или инструмента вроде *Phatch* (PPhoto BATCH Processor – пакетный процессор фото), имеющегося в большинстве репозиториях.

Фотографы, однако, как правило, являются визуалами, и им на радость создано множество GUI-программ, помогающих организовать изображения после съемок. Выделяются две из них, в основном благодаря их скорости и возможности просмотра RAW-файлов: это *Picasa* от Google и превосходная *Digikam*.

Picasa не является приложением открытого исходного кода, но эта программа бесплатна, и ее можно скачать с picasa.google.com.



com. Она очень проворна и делает большую работу, отслеживая, что находится на вашем жестком диске. Просто перетащите фотографии с карты памяти, и *Picasa* проиндексирует их таким образом, чтобы вы могли выполнять поиск по дате или названию файла, а затем прописывать тэги.

Digikam – просмотрщик фотографий по умолчанию для KDE, но прекрасно работает и в Gnome. По скорости он сравним с *Picasa*, но временная шкала на главном экране *Picasa* немного лучше, и вы можете перетаскивать и помечать фотографии из любого окна для того, чтобы реорганизовать их на вашем диске.

Обе программы предлагают встроенные RAW-конвертеры и редакторы изображений, хотя таковой в *Picasa* не вышел далеко за пределы небольшой поправки снимков с праздничной пьянки. А вот *Digikam* поставляется с мощным редактором *Showfoto*. Он не только даст вам контроль над настройками RAW, такими как корректировка выдержки и баланса белого: он умеет еще и обрабатывать 16-битные изображения и поддерживает профили управления цветом (см. врезку про калибровку).

Но и *Digikam* не лишен ограничений. С точки зрения инструментов, нет никакой очевидной поддержки модулей расширения, и ряд жизненно важных функций, таких как предустановленные кривые, резкость и локальный баланс белого, отсутствует в им-

» Привычный интерфейс *Darktable* вскрывает невероятные уровни сложности, когда дело доходит до настройки инструментов.

Лучшие 16-битные редакторы?

Хотя RAW-файлы вполне возможно распечатать, используя *Rawstudio*, *Photivo*, *Darktable* и иже с ними, ни одна из этих программ не имеет основных инструментов редактирования – вроде кисти клонирования для удаления следов пыли с фотографии – и вам потребуется другая программа, чтобы подправить ваши снимки перед их отправкой на принтер.

На данный момент, *GIMP* не поддерживает 16-битные изображения, и его следующая версия, 2.8, тоже не будет. Тогда остается два варианта: *CinePaint* или *Krita*.

CinePaint (www.cinepaint.org/more) основана на *GIMP*, но была разработана для киноиндустрии – для редакти-

рования кадров из кинофильмов. Исходный код ее открыт, но не очень хорошо поддерживается, и последний раз она обновлялась в 2008 году.

По части настройки уровней и кривых тона *Krita* в области редактирования фотографий уступает мощи *CinePaint* или любого из RAW-конвертеров, но имеет кисть клонирования и работает с 16-битными изображениями. Для большинства стилей фотографий, не требующих сложного и явного редактирования, это все, что нужно для завершающего шага в рабочем процессе, хотя в рабочем столе Gnome *Krita* может изрядно тормозить.



» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

портере RAW. Еще более критично, однако, то, что нужно пройти через три окна, чтобы перейти с экрана управления активами к редактированию RAW-файлов: это преступно много.

Разгон системы

Тенденция последних лет в проприетарном программном обеспечении – переход на единый интерфейс для всего, и именно это делают программы наподобие *Lightroom 3* или *Aperture*.

Интерфейсы у них очень похожи и основаны на практически забытом приложении под названием *RAWShooter* – пять лет назад Adobe купил его и включил в *Lightroom*. Слева – файловый браузер, а справа – панель регулировки параметров для проявления RAW-файлов. В центре – таблица цветов: панель, которая может показать контакт-лист (предпросмотр миниатюр) из определенной папки или библиотеки, или увеличить до полноэкранный режим манипуляции или просмотра двух изображений рядом друг с другом. В отличие от *Digikam*, здесь все делается на одном экране, что значительно ускоряет рабочий процесс.

Особенностью разработки программ такого рода является требование высокой эффективности по потреблению памяти. Плюс, и это важный момент, помимо применения корректировки по всей картинке последние версии этих программ также включают инструменты для редактирования слоев и удаления пятен. Иногда это означает, что 80–90 % кадров можно сразу отправлять на принтер, даже не открывая *Photoshop*-подобных программ.

Есть много инструментов, во многом похожих на *Lightroom*, *Aperture* и *Bibble*, но с открытым исходным кодом. *Darktable* (darktable.sourceforge.net) и *Rawstudio* (rawstudio.org) – мощные цифровые инструменты для «проявки» изображений, которые дают огромные возможности контроля. Оба можно построить из исходника или взять из PPA, и оба могут автоматически корректировать бочкообразные искажения и хроматические аберрации при помощи библиотек *Lensfun* (<http://developer.berlios.de/projects/lensfun/>).

Попробовать стоит оба: *Rawstudio* быстр и эффективен, но его встроенный браузер файлов ограничен, и нельзя сравнить изображения, поместив их рядом с друг другом. Здесь мало контроля над столь важными вещами, как настройка резкости, поэтому вам придется экспортировать TIFF во второй редактор. *Darktable*, с дру-



► *Rawstudio* прекрасно подойдет, если вам требуется только «проявить» ваши снимки, но для их окончательной отделки нужен и отдельный редактор.

гой стороны, имеет огромное количество опций и модулей расширения, но замедляется при обработке множества миниатюр для предпросмотра. Что более критично, ему не хватает глобальных пресетов для быстрого придания заданного «вида». А в рабочем процессе профессионала наличие ассортимента основных преобразований, выполняемых одним щелчком, жизненно важно.

И нам остается самый популярный среди редакторов открытого кода для RAW, *RawTherapee* (rawtherapee.com). Во многих стандартных репозиториях имеется стабильная версия *RawTherapee 2*, но третий релиз обкатывается уже продолжительное время, и мы бы настоятельно рекомендовали взять его из PPA-архивов или скомпилировать из исходного кода.

Заметно, что в мире открытого исходного кода *RawTherapee* ближе всех подобрался к *Bibble*, и он действительно очень хорош. Мало того, что он быстро создает библиотеку миниатюр изображений для предпросмотра и имеет все необходимые функции для работы со светом – он также поддерживает цветовые профили на протяжении всей работы, несколько мониторов и фильтрацию библиотеки по метаданным или тэгам. Плюс, каждая миниатюра для предпросмотра снабжается простым EXIF-довеском,

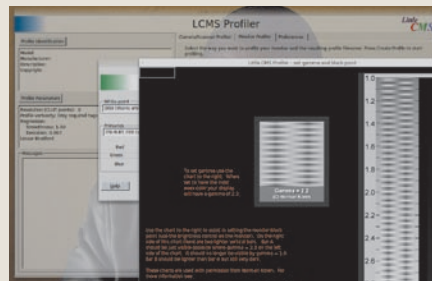
Калибровка монитора

Каждый монитор имеет свои уникальные свойства, когда дело доходит до отображения цвета на экране. Белое на одном дисплее может стать розовым на другом, а при печати снова выглядеть несколько иначе.

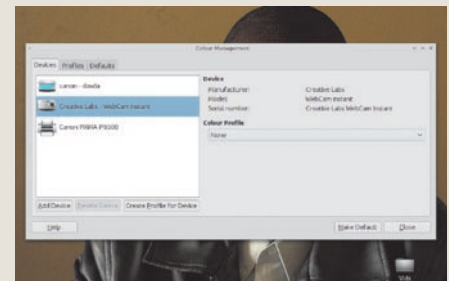
Профессионалы в обработке изображений выполняют калибровку своих мониторов с целью компенсации подобных различий, но в Linux издавна повелось считать это сложной задачей.



► Если ваш монитор имеет достаточно регуляторов настройки, воспользуйтесь отличным руководством по калибровке на www.lagom.nl, чтобы настроить параметры вручную и в большинстве случаев получить экран, достаточно близкий к стандартным цветам sRGB.



► Есть более эффективные способы профилирования. *LProf* (lprof.sourceforge.net) выполняет калибровку всего набора при помощи ссылок на целевой цвет. Они сохраняются в стандартных ICC-форматах, и должны быть загружены во всех используемых вами программах.



► Чуть проще в применении будет *Gnome Color Manager* (<http://projects.gnome.org/gnome-color-manager/index.html>). Это графический интерфейс для *Argyll*, проекта с открытым кодом по созданию драйверов для инструментов калибровки оборудования типа Datacolor Spyder.

► Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

наглядно показывающим условия съемки – например, фокусное расстояние и значение ISO. Это удивительно редкостная функция в столь легкой для просмотра форме.

Наша единственная жалоба – на избыток параметров для настройки. Вы можете запутаться в настройках, проходя через различные меню, и значения нередко дублируются.

Упорядочив свои фотографии, переходите к ним через левую панель *RawTherapee*. Оказавшись на месте, вы можете скрыть эту панель с помощью значка стрелки в верхней части экрана.

Теперь, видя перед собой эскизы, вы можете отобрать те, что на ваш взгляд, стоит «проявить». *RawTherapee* дает навесить каждому фото до пяти звезд; применима и двоичная система – «со звездой или в корзине». По мере прохождения через вашу выборку допускается удалять звезды с фотографий, которые не так хороши, как вам сгоряча показалось. Фильтр на правой панели позволяет дальнейшую сортировку.

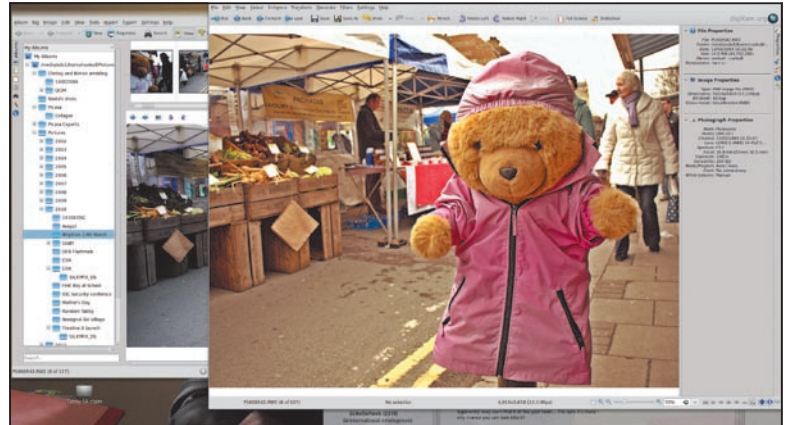
Порядок применения изменений к RAW-файлам значения не имеет – сохраняются только конечные величины. Хорошей стратегией, однако, будет начать с выдержки и контраста. Нескорректированные RAW-файлы выглядят более темными, плоскими и слегка расфокусированными по сравнению с JPG на выходе с камеры. Не расстраивайтесь, это всего лишь вид: конвертор и навыки фотографа заставят их буквально выпрыгивать с листа. Почему это происходит на белой бумаге, отлично объяснено на сайте компании Adobe: <http://bit.ly/jLOvWK>.

Благодаря дополнительной глубине цвета, выдержку в RAW-файле обычно можно изменить на шаг или два в любом направлении, чтобы затемнить или осветлить изображение. С помощью этого ползунка в паре с контрастом вы получите темную форму графика гистограммы в верхней части правой колонки, распределенную максимально равномерно.

Далее, я обычно корректирую баланс белого, щелкнув в области, которая, как я знаю, должна быть белой. Вручную установить ползунок непросто, поэтому старайтесь держаться этой тактики – или воспользуйтесь для компенсации эффектов освещения одним из заданных оттенков. Потом я обычно добавляю немного насыщенности. Будьте осторожны: большинству из нас нравится придавать энергии своим снимкам, но здесь легко переборщить.

Самое сложное в редактировании RAW-изображений – это резкость и снижение уровня шума. Всегда, всегда работайте с масштабным соотношением 1:1, чтобы видеть фотографию в реальном размере. Изменения, вносимые здесь, в другом масштабе могут выглядеть совсем иначе, и нередко катастрофически.

В зависимости от марки и модели вашей цифровой зеркалки вам, возможно, потребуется завесить Unsharp Mask до 200,



а то и больше, чтобы получить действительно четкие края. Что касается шумоподавления – новые алгоритмы *RawTherapee Impulse* очень хороши, но помните, что при размерах изображений, полученных современными камерами, небольшой шумок гораздо более приемлем по сравнению с потерями в детализации. Лучше перебдеть.

Желая получить наилучшие результаты, используйте Контраст по уровням вместо Unsharp Mask. *Rawstudio* по-прежнему отстает в этой области от проприетарного программного обеспечения: несколько кадров, которые я отснял в темных, сложных местах (например, свадьбы в церкви) были неремонтопригодны в *RawTherapee*, но после обработки в *Bibble* стали героями дня.

» *Digikam* способен стать идеальным конечным инструментом для профессионалов, работающих в Linux. Но пока он немного тормозит при множестве фотографий.

Справляется ли Linux с работой?

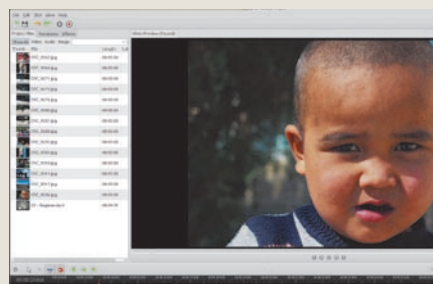
Одна из программ, доступных в Linux, которая умеет делать все, на что способны *Lightroom* и *Aperture*, и направлена на профессиональное использование – проприетарная *Bibble* (bibblelabs.com). Мы не предлагаем вам отказаться от философии FOSS и пойти и купить ее – хотя она стоит затрат; мы здесь продемонстрировали, что Linux ничто не препятствует быть столь же способной к высококачественной обработке фотографии, как и другие системы. Инструменты FOSS отнюдь не далеко позади.

Если когда-нибудь кто-нибудь усомнится в способностях вашего любимого СПО для фотографии, просто укажите им на замечательный обзор Photopic Sky Survey (skysurvey.org). Эта 5000-мегапиксельная фотография ночного неба над Землей заняла год долгих и любовных трудов шивания вместе 37440 отдельных изображений. Она угодила в заголовки журнала *New Scientist* и многих популярных газет. И все это было сделано в *GIMP*. Так-то. **LXF**

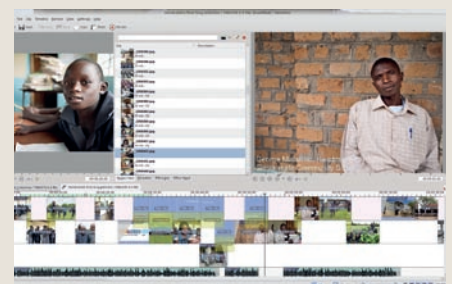
Создание слайд-шоу



» Слайд-шоу могут наводить на мысль о соседском поздраве фотоальбомов, но в руках профессионалов это один из лучших форматов для рассказа фотоисторий. Не верите нам? Посмотрите duckrabbitt.info.



» Ключ к хорошему фотофильму – возможность контроля времени показа каждого изображения и используемые эффекты перехода. Лучше использовать нелинейный видеоредактор типа *OpenShot*, а не специальный инструмент для фото.



» Для применения переходов типа «эффект Кена Бернса» – где камера панорамирует статичное изображение – вам нужно нечто более мощное. Видеоредактор *Kdenlive* может показаться перебором, но это единственный способ сделать все правильно.

Майк Сондерс ведет

ТРЕНИНГИ LINUX



Linux Professional Institute

Наш эксперт

Майк Сондерс пишет о Linux более десяти лет, и установил больше дистрибутивов, чем съел горячих обедов.

Часть 7: Учитесь управлять процессами как профи и овладейте редактором *Vi*, который славен своим минимализмом.

Mike



Наша серия про LPI близится к концу – последняя статья выйдет в следующем месяце; и пора обратиться к сложностям, встречающимся на пути системного администратора.

Для начала бросим взгляд на процессы и узнаем, как управляться с ними по своему вкусу. Нет ничего хуже, чем блуждающий процесс, который удаляет системные файлы, оставляя вас беспомощным, и мы рассмотрим, как решается эта проблема. Мы также взглянем на файловые системы не с точки зрения содержимого, как раньше, а узнаем, как отформатировать разделы под файло-

вые системы новых типов и выполнить их проверку на случай, если что-то пойдет не так.

Наконец, мы исследуем редактор *Vi*, поставляемый почти с каждым дистрибутивом Linux и Unix; он считается трудным в освоении, но для разобравшихся в его основах становится просто даром Божиим.

В следующем номере мы приведем подробный список учебных вопросов для LPI, поэтому, когда вы закончите с этим руководством, у вас будет несколько недель, чтобы освежить знания. Удачи!

Раздел 1: Управление процессами

Представьте, что вы, сидя дома, размечтались о чашечке чая. Как человек предприимчивый, вы просите ближайшего члена семьи ее вам приготовить. Вышеупомянутый член рысит на кухню; но, обернувшись в кресле, вы видите, что он/она вместо чая заваривает кофе. Вы восклицаете «Черт побери, я же просил чаю!» или нечто подобное. А как это звучит в терминах компьютера? Что происходит, если вы запустили программу и хотите остановить ее или изменить способ ее работы?

На простейшем уровне, выкрику «стоять!» соответствует нажатие Ctrl+C. Попробуйте эту комбинацию клавиш с командой, генерирующей большой объем вывода, вроде `ls -R` / для вывода содержимого корневого каталога и всех его подкаталогов. Когда она примется изрыгать тысячи строк на терминал, можно нажать Ctrl+C и остановить ее на полпути. Команда остановлена; больше ничего не делается. Это крайне важно на случай, если сдуру введено что-то неправильное и надо срочно пресечь команду, пока она не нанесла ущерб.

Однако бывают и другие варианты. Что если вы просто хотите отложить выполнение программы на потом? Например, вы ввели команду `man gcc` (чтобы прочесть страницу справки по компилятору GNU C Compiler), прокрутили текст вниз и увидели что-то интересное – вам хочется это опробовать, не теряя положения в тексте. Нажмите Ctrl+Z, и страница просмотра справки исчезнет в фоне, оставив вас в командной строке. Сделайте то, что хотели, и наберите `fg` (от foreground – передний план) – вы вернетесь в то же место страницы справки, где и были.

Программу можно запустить в невидимом режиме (приостановив), чтобы вы могли вернуться к ней, когда созреете. Для этого добавьте к имени команды символ амперсанда (&) следующим образом:

```
man df &
```

Здесь мы открываем для просмотра страницу справки для команды `df` (она отображает объем свободного места на диске), но делаем это в фоновом режиме. Экран отреагирует строкой

» **Месяц назад** Продвинутые трюки и приемы командной строки.

Рад видеть тебя, видеть тебя, nice

По умолчанию права доступа к ресурсам у всех процессов одинаковы. Если процессы А и В оба потребуют время процессора, планировщик ядра Linux разделит это время между ними поровну. А это не всегда желательно, особенно если много процессов выполняется в фоновом режиме. Например, может по расписанию включиться задание cron на сжатие старых архивных файлов на домашнем компьютере – если пользователь делает нечто срочное, вы вряд ли захотите, чтобы из-за этого он внезапно лишился 50 % вычислительной мощности.

Борется с такими ситуациями система приоритетов. У каждого процесса есть значение **nice** [англ. любезность], определяющее его послушание операционной системе. Наибольшее значение **nice** равно 19; по умолчанию оно равняется нулю и может доходить до -20 (наибольший приоритет). Например, для запуска программы с наименьшим приоритетом воспользуйтесь командой

```
nice -n 19 programname
```

Программа запустится и, если в системе больше ничего не происходит, завершится в отведенный

срок. Но при наличии других процессов система сначала отработает с теми. Для задания значений **nice** ниже нуля вы должны иметь права root:

```
sudo nice -n -10 programname
```

Это касается многопользовательских систем, где администратор отдает приоритет своим задачам (иначе все остальные называли бы максимальные приоритеты своим процессам!). Значение **nice** процесса можно изменить командой **renice** – более подробную информацию см. на ее странице справки – а значения **nice** выводит команда **top**.

[1] 3192

В скобках показан порядковый номер фоновой программы; второе число – идентификатор процесса (мы перейдем к ним через минуту). Теперь занимайтесь делом, а когда нужно будет пристроиться к запущенной программе, просто введите **fg**.

Эта система становится особенно удобной при манипуляции несколькими программами. Например, введите

```
nano &
```

```
man df &
```

Здесь мы запустили две программы в фоновом режиме. Если ввести **jobs**, мы получим их список с номерами и командами их запуска. Можно восстановить выполнение программы по ее номеру: например, **fg 1** переключит нас в редактор *Nano*, а **fg 2** – в программу просмотра страницы справки.

Теперь – к процессам. По сути, процесс – это экземпляр выполняемой программы. Большинство простых программ создают единственный процесс: самое себя. Более сложные программы, такие как рабочий стол, запускают множество процессов – демоны мониторинга файлов, оконные менеджеры и т.д. Это помогает обслуживать систему (представьте, что весь KDE был бы в одном огромном исполняемом файле, и при выходе из строя одного компонента рухнуло бы все), а также позволяет делать всякие удобные вещи.

Чтобы вывести список процессов, наберите **ps**. Вывод команд вряд ли вас вдохновит – скорее всего это будет лишь пара строк: команда выведет только процессы, запущенные текущим пользователем. Для просмотра всех процессов, выполняющихся на компьютере, введите **ps ax**. Обычно их список очень длинен, и лучше переправить его через канал команде **less**, как мы рассказывали в прошлом месяце:

```
ps ax | less
```

Что именно вы увидите, зависит от конкретной сборки дистрибутива Linux и запущенных в данный момент программ, но вот вам пример:

```
2972 pts/0 Ss 0:00 bash
```

Здесь **2972** – идентификатор процесса (PID). У каждого процесса есть уникальный идентификатор, начиная с номера 1, соответствующего программе **/sbin/init**, которую ядро запускает при загрузке. Следующий фрагмент **pts/0** показывает, с какого виртуального терминала была запущена команда – если здесь вы видите знак вопроса, то процесс был запущен вне терминала: например, ядром или скриптом загрузки. **Ss** означает, что процесс спит (не производит активной работы), затем идет индикатор времени, показывающий, сколько процессорного времени процесс получил на данный момент, а за ним следует команда, запустившая процесс.

» Вот вывод команды **ps ax**, показывающий все процессы, запущенные в системе.

Еще один способ получения списка процессов (по умолчанию отсортированного по потреблению процессора) – команда **top**. Это интерактивная программа – она обновляется каждые несколько секунд, отображая в верхней части списка задачи, наиболее активно использующие процессор. Столбцы снабжены заголовками (в черном прямоугольнике), и вы можете определить идентификатор процесса, пользователя, который его запустил, и т.д. Обратите внимание на различные столбцы с информацией об использовании памяти: самый важный из них – RES (resident – резидентная часть), в нем показан объем реальной памяти, который процесс использует в данный момент. Для выхода из **top** нажмите Q.

Итак, предположим, что вы удачно запустили программу, но внезапно она дала сбой: застряла в цикле и съедает все процессорное время. Завершить ее, нажав Ctrl+C, нельзя – вы запустили программу не из терминала, а из оконного менеджера. Что делать? Первая возможность – определить идентификатор с помощью описанных выше методов и ввести команду

```
kill <pid>
```

Замените **<pid>** идентификатором процесса. Хотя эта команда и намекает на убийство, но по умолчанию ведет себя довольно мягко. От себя лично **kill** отправляет процессу дружелюбное сообщение «Не хотите ли завершиться?», получив которое, процесс может предпринять необходимые действия (например,

Скорая помощь

Комбинации клавиш для остановки и приостановки процессов, такие как Ctrl+C и Ctrl+Z, работают в большинстве случаев, но не всегда. Обработку этих клавиш можно переназначить программно, задав им другой функционал или сделав так, чтобы сотрудники не забрасывали важные программы ради игры в *Nethack*!

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Скорая помощь

Краткий обзор производительности вашей системы можно получить по команде **uptime**. Она показывает, сколько времени истекло с момента последней (пере)загрузки, сколько пользователей в системе, и среднюю нагрузку за последние периоды времени.

удалить временные файлы перед завершением). Иногда это остановит процесс, но если у процесса есть собственный обработчик сигнала **kill** и по причине сбоя он уже ни на что не способен – вы в тупике. Тут **kill** начинает оправдывать свое название. Введите

```
kill -9 <pid>
```

Команда даже не вникает, что там творится с программой – она просто останавливает ее немедленно. Если процесс в этот момент записывал данные в файл, результаты могут оказаться весьма беспорядочными, поэтому данный способ следует применять весьма осторожно и только тогда, когда исчерпаны другие возможности.

Иногда бывает несколько процессов с одним и тем же именем, или вам просто лень искать PID процесса. В этом случае пригодится команда **killall**. Например, пусть вы скомпилировали новейший модуль *Apache*, добавили его в *Apache...* и ваш web-сервер пошел наперекосяк, но не дает себя остановить обычными скриптами,

причем запущено множество экземпляров **apache**. Попробуйте такую команду:

```
killall -9 apache
```

Другой полезный сигнал, который не является разрушительным, а лишь велит программе перезапуститься или перечитать свои файлы настройки – **SIGHUP**, названный по выражению «hanging up» [вешать трубку, разъединяться] из эпохи телефонных модемов. Многие программы игнорируют этот сигнал, но он прекрасно работает с рядом фоновых демонов, таких как серверы:

```
killall -HUP sendmail
```

Эта команда велит всем выполняющимся процессам **sendmail** на секунду приостановиться, перечитать файлы настройки и продолжить работу. Это очень удобно, когда вы хотите быстро изменить файл настройки, не останавливая всю программу. Если по какой-то причине вы не хотите, чтобы этот сигнал обрабатывался, воспользуйтесь **nohup** для его отключения – более подробную информацию см. в **man nohup**.

➤ Команда **top** выводит список процессов, отсортированный по потреблению процессора.

Раздел 2: Создание новых файловых систем

Перейдем к сложной теме, которой вам как системному администратору обычно касаться не придется – файловым систе-

мам и разбиению на разделы. По крайней мере, такое случается не каждый день. В большинстве случаев настройки разделов задаются при установке и сохраняются месяцами или даже годами. Графические утилиты вроде *GParted*, используемые во многих установщиках, значительно упрощают этот процесс, но на случай внештатных ситуаций важно иметь представление об утилитах командной строки.

Для начала напомним: жесткий диск обычно делится на несколько разделов. Например, у вас может быть раздел с **Windows** и раздел с **Linux**. Большинство систем **Linux** располагаются на двух или более разделах: разделах данных (таких как **/** и **/home**) и разделе подкачки для виртуальной памяти. Данные на этих разделах должны храниться в определенном порядке, или формате, и исторически в **Linux** это был формат файловой системы **ext2**. Потом появилась **ext3** (то же, что и **ext2**, но с добавлением журналирования), а теперь – **ext4**. Другие типы разделов из мира **Linux** и **UNIX** включают **XFS** и **ReiserFS**. В большинстве систем **Windows** используется **NTFS**, но на внешних носителях, таких как USB-брелки, обычно используется **FAT32** (также известная в **Linux** как **VFAT**).

```
top - 19:21:43 up 2:47, 2 users, load average: 0.26, 0.33, 0.32
Tasks: 114 total, 1 running, 112 sleeping, 0 stopped, 1 zombie
Cpu(s): 1.0%us, 1.3%sy, 0.0%ni, 96.7%id, 0.0%wa, 0.0%hi, 1.0%si, 0.0%st
Mem: 515316k total, 461756k used, 53560k free, 36100k buffers
Swap: 1048568k total, 0k used, 1048568k free, 281492k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2720	root	15	0	52720	10m	5636	S	1.0	2.1	0:11.45	Xorg
2323	root	18	0	1976	636	556	S	0.3	0.1	0:13.32	hald-addon-stor
2818	mike	15	0	33704	7880	6524	S	0.3	1.5	0:00.62	gnome-settings-
2838	mike	15	0	16596	7304	5776	S	0.3	1.4	0:00.43	metacity
2901	mike	15	0	44884	6152	4648	S	0.3	1.2	0:02.23	gnome-power-man
2907	mike	17	0	18208	2424	1984	S	0.3	0.5	0:00.99	escd
3398	mike	15	0	39992	12m	9080	S	0.3	2.5	0:03.30	gnome-terminal
1	root	15	0	2072	656	564	S	0.0	0.1	0:01.53	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.03	ksftirqd/0
4	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
5	root	10	-5	0	0	0	S	0.0	0.0	0:01.64	events/0
6	root	10	-5	0	0	0	S	0.0	0.0	0:03.16	khelper

Проверка целостности файловой системы

Современные файловые системы **Linux**, такие как **ext4** (система по умолчанию в большинстве современных дистрибутивов), безотказны и надежны. Они не сотворят чудес при сбое электросети, но могут попытаться оставить файловую систему в относительно порядке, чтобы вы не потеряли все данные. Однако все на свете уязвимо, и если у вас проблема с «железом», можно заподозрить, что сломался жесткий диск. Как решил бы вопрос администратор?

Во-первых, наберите **dmesg** и посмотрите, нет ли чего веселенького в файлах журналов: сообщений об испорченных данных или секторах и т. д. Если вы узрели нечто подобное, сразу же вставьте в компьютер съемный USB-диск и скопируйте все важные данные, потому что нельзя знать заранее, когда диск окончательно рухнет. Затем по команде **df -h** определите объем свободного места на диске; если он го-

раздо меньше, чем вы ожидали, что-то явно не так. Для просмотра дискового пространства в каталогах пользуйтесь **du -h**.

Следующий шаг – проверка файловой системы. Перезагрузите свою систему в однопользовательский режим (способ этой перезагрузки зависит от дистрибутива, но обычно включает изменение строки конфигурации ядра в загрузчике и добавление в конец **S**.) Войдя в терминал, введите:

```
/sbin/fsck device-name
```

Измените **device-name** на файл устройства основного корневого раздела, например, **/dev/sda1**. Так же можно проверить и другие разделы. **fsck**, по сути, внешняя оболочка для утилит проверки разных файловых систем и в большинстве компьютеров с **Linux** запускает **/sbin/e2fsck**, которая работает с файловыми системами **ext2/3/4**. Если проверка обнару-

жит проблемы, **fsck** спросит вас, что нужно сделать. После проверки можно запустить **/sbin/dumpe2fs** с именем устройства и получить более подробную информацию о разделе – она пригодится для сообщения о проблеме на интернет-форуме.

Вы можете заметить, что во многих дистрибутивах **Linux** проверки файловой системы с **fsck** автоматически запускаются после каждых 30 загрузок, или каждых 100 дней, или и того, и другого. Изменяет этот механизм утилита **tune2fs** с опциями **-c** и **-C**. Она также может задать действия ядра при обнаружении ошибок и много чего еще. Имеет смысл прочесть ее страницу справки, особенно информацию о первых пяти опциях. Похожая, хотя и глубоко техническая утилита – **debugfs**, но чтобы пользоваться ею продуктивно, нужно отлично знать внутреннее устройство файловой системы.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Основная утилита разбиения диска на разделы с командной строки – *fdisk*. В качестве ее аргумента нужно указать файл устройства для вашего жесткого диска следующим образом:

```
/sbin/fdisk /dev/sda
```

(Учтите, что запускается эта команда от имени *root*; а если вы не уверены в том, какой файл устройства соответствует жесткому диску, загляните в вывод *dmesg*.) Также обратите внимание, что здесь указано */dev/sda*, а не */dev/sda1*: последняя цифра – номер конкретного раздела, а нам необходим весь диск. В большинстве случаев *sda* – первый жесткий диск, *sdb* – второй, и т.д.

fdisk – довольно «голая» программа, без меню или мастеров для автоматизации действий. Введите *p* для вывода списка разделов своего жесткого диска и *m* для вывода справки. Вы увидите команды для удаления разделов, создания новых разде-

лов и т.д. Все производимые изменения не записываются на диск до тех пор, пока вы не нажмете *w*. В некоторых дистрибутивах есть *cfdisk*, версия *fdisk* на основе *curses*, где все немного удобнее: есть простые меню, а перемещаются по ним клавишами управления курсором.

Создав раздел, отформатируйте его. Тут в дело вступают утилиты *mkfs*. Наберите */sbin/mkfs* и нажмите *Tab* для вывода возможных вариантов – вы увидите *mkfs.ext3* (для большинства разделов Linux), *mkfs.vfat* (для разделов FAT32) и другие. Для форматирования раздела просто укажите его файл устройства:

```
/sbin/mkfs.ext3 /dev/sda2
```

Для разделов подкачки пользуйтесь командой *mkswap*. Затем можно активировать и деактивировать раздел подкачки командами *swapon* и *swapoff*.



Чтобы узнать объем доступной памяти, введите *free -m*. Команда отображает статистику в мегабайтах. Первая строка может шокировать вас и заставить подумать, что памяти почти не осталось, хотя у вас работает один *Fluxbox*. Но данная цифра включает кэш на дисках, поэтому смотрите на вторую строку: *-/+buffers/cache*.

Раздел 3: Краткий обзор редактора Vi

Наконец, в этом месяце мы бегло оглядим *Vi*, «визуальный» редактор. А не нелепо ли описывать редактор как «визуальный» – разве не все они такие? Но в семидесятые годы, когда *Vi* разрабатывался для операционных систем Unix, кое-кто еще пользовался телетайпами...

Идея полноэкранного редактора была достаточно новаторской, так как люди привыкли работать с отдельными строками текстового файла. Редактор *Vi* – очень сжатый и примитивный, но благодаря его низким требованиям устанавливается почти на каждый компьютер с Unix. В мире Linux большинство дистрибутивов содержат *Vim* (*Vi Improved*), гораздо более продвинутую версию редактора.

Начинаем

Для запуска редактора наберите *vi filename.txt*. Прежде чем колотить по клавишам, обратите внимание, что *Vi* работает в двух режимах: обычном (для команд) и вставки (для редактирования текста). Это отличает его от большинства других редакторов, в которых сразу набирается текст.

В *Vi* для ввода текста в текущей позиции курсора нужно нажать *i*, после чего вы сможете набрать то, что хотите. Закончив, нажмите *Esc* для возврата в обычный режим ввода команд.

Команд в *Vi* немало, и если вы хотите стать гуру *Vi*, то специальных книг про него хватает. Но вот самые основы: в обычном режиме ввод *dd* удаляет строку, *yy* копирует строку в буфер обмена, а *p* вставляет эту строку обратно в текст.

Для выполнения некоторых операций сначала нужно набрать двоеточие. Например, *:w* записывает файл на диск, а *:q* завершает работу редактора. (Если вы что-то переделали, не сохранили изменения и попытались выйти, *Vi* может пожаловаться – выйти без сохранения данных можно командой *:q!*.) Действия можно объединять: например, команда сохранения и выхода – *:wq*.

Многие люди считают *Vi* и *Vim* дико неудобными в работе и предпочитают редакторы без командного режима, типа *Emacs* или *Nano*. Другие, наоборот, обожают первые два из вышеупомянутых за минимализм и ненавидят склонность двух последних неумеренно злоупотреблять клавишей *Ctrl*. Эта война пылает постоянно, но, независимо от ее исхода, все хорошие администраторы должны иметь базовые навыки работы в *Vi*, так как он почти наверняка будет на любом компьютере, который вам попадется. **LXF**

```
VIM - Vi IMPROVED

version 7.0.237
by Bram Moolenaar et al.
Modified by <bugzilla@redhat.com>
Vim is open source and freely distributable

Help poor children in Uganda!
type :help iccf<Enter>      for information

type :q<Enter>              to exit
type :help<Enter> or <F1>   for on-line help
type :help version7<Enter> for version info
```

» В отличие от обычного *Vi*, *Vim* при запуске без имени файла в качестве аргумента любезно показывает кое-какие справочные инструкции.

Проверьте себя!

По мере продвижения по этой серии вы можете захотеть оценить свои знания: ведь вас ждет сертификация LPI, и вы должны уметь применять свои знания, не заглядывая в руководства. Итак, убедитесь, что вы прочли, усвоили и попробовали на практике весь материал этого руководства, и посмотрите, сможете ли вы ответить на следующие вопросы.

- 1 Как запустить команду *exim -q* в фоновом режиме?
- 2 Есть несколько программ, запущенных в фоновом режиме. Как вывести их список?
- 3 Как получить список процессов?
- 4 *Exim* стал чудить, и вам нужно полностью завершить все запущенные экземпляры программы. Какой командой это делается?
- 5 Вы только что создали новый раздел */dev/sda2* и хотите отформатировать его в FAT32. Как?
- 6 Напишите команду для запуска *myprog* с самым низким приоритетом.

```
1 - exim -q - 2 - 3 - ps ax 4 - killall -9 exim 5 - /sbin/mkfs.vfat /dev/sda2
```

» **Через месяц** Финальные аккорды и парад контрольных вопросов.

Рудим облаком:

Горечь утраты флэшки более испытывать не придется. Эндрю Грегори покажет, как присмотреть за своими документами, употребив мощь облаков.



Ежегодно теряются тысячи часов работы, потому что людям никак не вспомнить, где они сохранили свои файлы. На компьютере в офисе? На корпоративном ноутбуке? Может быть, записали на DVD и сунули его в коробку из-под Lunugd Skynugd? Или они лежат на флэшке, оставшейся в кармане других штанов? Раньше эту проблему обходили, посылая письма на учетную запись web-почты, но тогда дублируются файлы и зря тратятся драгоценные мегабайты вашего ящика. Решение этой задачи и многих других – синхронизация данных в облаке.

Прежде чем двигаться дальше, отметим, что хранение данных в облаке – не для всех, и не для любых типов данных. Доверять важные данные облаку неумно, а если вас волнует конфиденциальность, вообще откажитесь от этого – хранение файлов на чужих серверах быстрое, простое и удобное, но вы препоручаете контроль над вашими файлами организациям, которые не на 100 % разделяют ваши соображения касательно безопасности и конфиденциальности. Конечно, преимущества хранения данных в облаке можно обрести и без этих ловушек, создав свое облако – см. соответствующий учебник Криса Брауна в следующем номере. Если же вы гонитесь только за экономией времени, читайте дальше.



Наш эксперт

Эндрю Грегори уже шесть лет пользовался всяческими сборками Linux, но время от времени устраивает поломки, просто для хохмы.

OO2GDocs

Если у вас есть учетная запись Gmail, вы, скорее всего, знакомы с *Google Docs* — бесплатным текстовым онлайн-процессором и электронными таблицами. *Google Docs* позволяет поработать на любом компьютере, подключенном к Интернету, отключиться, а потом забрать свою работу, не таская с собой физическую копию своих данных. Однако непрекращающаяся популярность настольных текстовых процессоров типа *AbiWord* и *LibreOffice* доказывает, что людям еще не совсем уютно вверять свои данные облаку. То есть если вы хотите пользоваться удобствами

Google Docs, не жертвуя полным контролем над своими данными, вам нужна некая синхронизация вашего текстового процессора с онлайн-сервисом.

Синхронно с LibreOffice

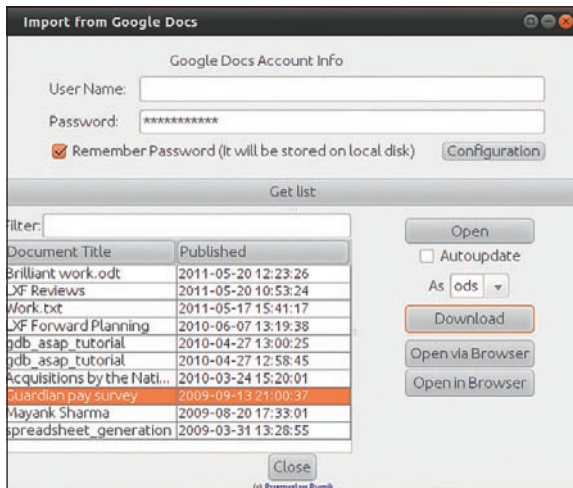
Вот вам и почва для *OO2GDocs*, модуля расширения для *LibreOffice*, написанного, как подсказывает название, в те времена, когда офисным пакетом номер один в Linux был *OpenOffice.org*. Для его установки скачайте файл расширения с <http://extensions.services.openoffice.org/project/ooo2gd>. В *LibreOffice*, перейдите в Tools > Extension Manager, и откроется Extension Manager; затем нажмите Add. Появится файловый диалог; через него можно перейти туда, где вы сохранили файл OXT (наш мы записали в папку **Downloads**). Нажмите Open, и расширение станет видимым в менеджере расширений. Оно станет доступным после перезапуска, и кусочек GUI, где отображены его функции, будет появляться в левом верхнем углу, пока вы не выберете для него место на панели инструментов *LibreOffice*. Мы выбрали самую правую часть нижнего ряда в панели инструментов – просто потому, что это самое свободное место в довольно скучном интерфейсе *LibreOffice*.

В идеальном мире этот модуль производил бы загрузку прямо во время вашей работы, но он пока еще не настолько умен, и сначала вам придется ввести черновик вашей новаторской диссертации, или другого шедевра, над чем вы там работаете, и сохранить его в вашей файловой системе. После этого нажмите на значок Export to Google Docs, расположенный там, где вы его поместили



OO2GDocs легко найти в панели инструментов LibreOffice — она там, где вы ее оставили.

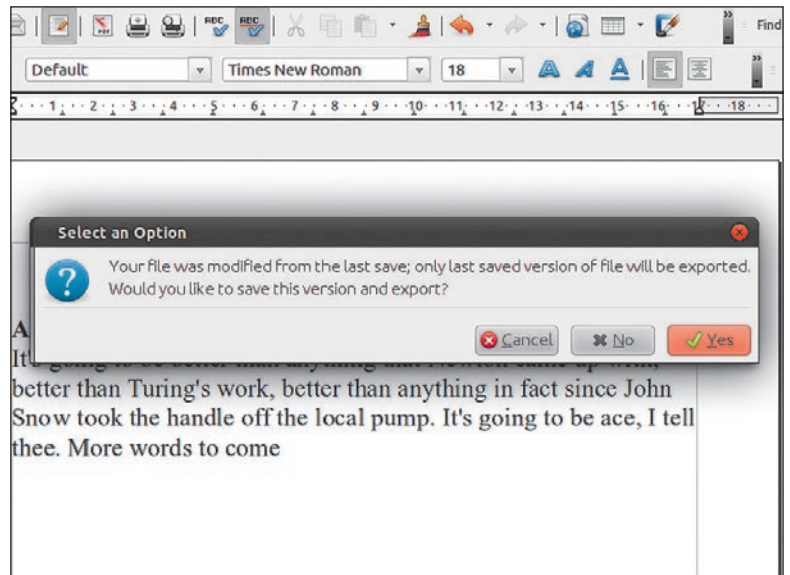
Работаем везде



► Из диалога Import From Google Docs можно скачать свою работу с Google Docs, посмотреть время последнего изменения файла и выбрать формат для файла.

(он отобразится вместе с зеленой стрелкой, указывающей на документ), и введите информацию о вашей учетной записи Google. В диалоге есть пара вариантов: Remember Password говорит сам за себя, как и Autoupdate. Просто поставьте здесь галочку, и каждое последующее сохранение будет автоматически подгружаться в Google Docs за какую-то пару секунд.

Процесс синхронизации в обратную сторону не действует: нельзя, поработав над документом в Google Docs, экспортировать его в LibreOffice. Но при желании заняться этим документом на другом компьютере достаточно будет нажать опции расширения рядом со значком загрузки и выбрать Import From Google



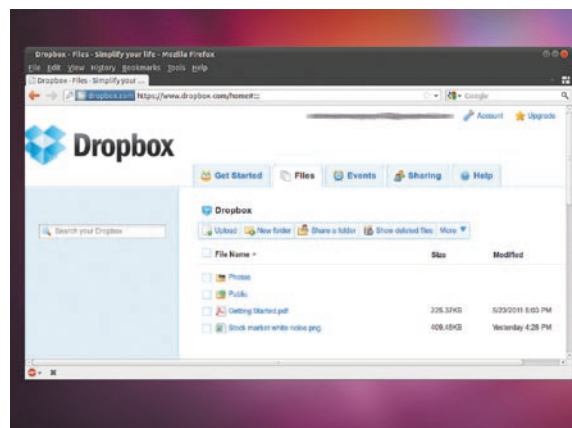
Docs. В открывшемся диалоге у вас, как и ранее, запросят данные об учетной записи Google; потом нажмите Get List, и отобразится список документов, доступных для скачивания с Google Docs.

Главный недостаток такого подхода – само расширение довольно ненадежное. На наших компьютерах (один под управлением Ubuntu 11.04 с Unity, а другой – с Linux Mint 10) оно то не откликалось и вызывало сбой, то работало как часы. 002GDocs написан на Java и даже при идеальной работе съедает ощутимый кус системных ресурсов; прибавьте это к собственным аппетитам LibreOffice, и у вас будет решение, пригодное разве что для новейшего компьютера, некритичного для производства.

► 002GDocs не может сохранять файл за вас до загрузки в облако, но как минимум предупредит, что вы забыли сохранить.

Dropbox

Более элегантное решение, хотя и не со всякой программой интегрированное, дает Dropbox. Он работает как удаленное хранилище, а при установленном клиенте Dropbox все выглядит как действия в локальном каталоге – примерно как в Konqueror, бывшем файловом менеджере KDE, пытались стереть грань между локальными файлами и файлами в сети. Dropbox доступен как RPM для Fedora и Deb для Ubuntu, или как архив tar.bz2 для любителей лично компилировать приложения. Установив его через менеджер пакетов или через www.dropbox.com, зарегистрируйте учетную запись (см. пошаговое руководство на следующей странице) и примите условия. Самое тягостное из них – то, что «Dropbox оставляет за собой право в любое время прекратить бесплатную учетную запись, с уведомлением или без». Ик! Среди условий затесано, что после 90-дневной неактивности бесплатной учетной записи Dropbox удалит ваши файлы без предупреждения. Мило, нечего сказать...



► Клиент Dropbox – более внятный способ загрузки файлов, но web-интерфейс полезен, если вы работаете на компьютере без этого клиента.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Следующим решением будет – принимать ли установку по умолчанию или задать настройки самим. По умолчанию Dropbox создаст папку для документов общего пользования в вашем домашнем каталоге и синхронизирует все документы с этой папкой, включая синхронизацию расширенных атрибутов файла: будет синхронизировано не только содержимое файла, но и его метаданные, то есть можно производить эффективный поиск и на других компьютерах, а не только на том, где файл создан.

Если клиент Dropbox установлен (например) на вашем домашнем и рабочем компьютере, «облачная» часть распределенного хранилища становится полностью невидимой. Просто откройте папку Dropbox в вашем файловом менеджере (или

с помощью значка, возникшего на панели инструментов), и там вы увидите все файлы, загруженные в учетную запись Dropbox. Единственный намек на посредничество Интернета – то, что при перетаскивании файла в папку Dropbox вы видите вращающийся синий значок, указывающий, что идет загрузка файла. В зависимости от скорости подключения это займет несколько секунд, после чего значок сменится на зеленую галочку – сигнал завершения загрузки.

Вот так все просто. Мало того, даже работая за заблокированным компьютером, где вам не дозволено устанавливать свои программы, вы все равно получите доступ к своим файлам на Dropbox через web-интерфейс www.dropbox.com.



Шаг за шагом: Установка Dropbox



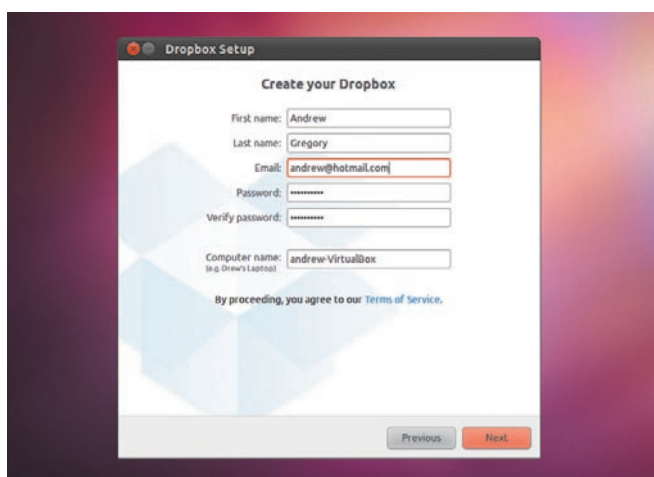
1 Демон, но не вольный

Увы, код демона *Dropbox* закрыт. Те, кто носит на голове кастрюлю, страшая радиочастот, могут думать, что он набит перехватчиками клавиш и прочим вредоносным ПО, но вряд ли это так.



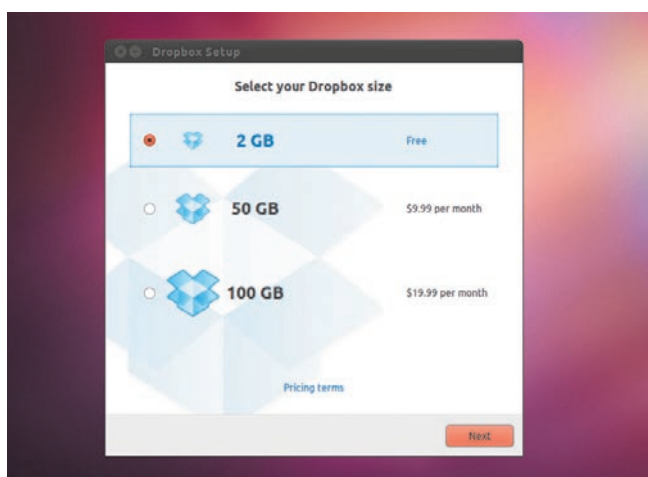
2 Создать учетную запись Dropbox

После установки программы *Dropbox Nautilus* и демона *Dropbox* вас попросят создать учетную запись.



3 Установка

Введите обычную информацию, обдумайте надежный пароль, прочтите условия и нажмите Next.



4 Выбор размера

Вам предложат на выбор разные объемы для учетной записи – 2 Гб бесплатно; 50 Гб за \$9,99 в месяц; или 100 Гб, это обойдется в \$19,99 в месяц.

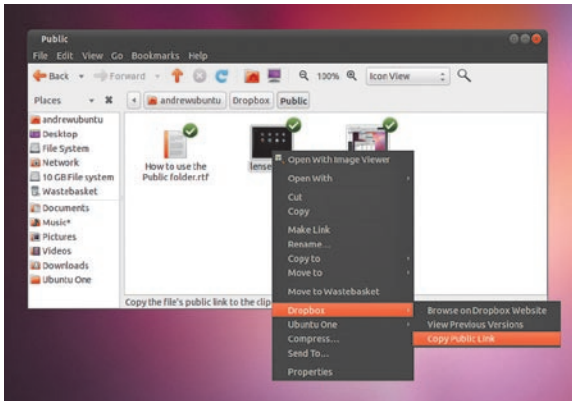
» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

А если вы хотите поделиться чем-либо с другом, не имеющим учетной записи и не желающим ее заводить (возможно, из боязни злодеяний файла с закрытым кодом), вы можете выдать в доступ документы в папке **Public**, расположенной в главной папке Dropbox.

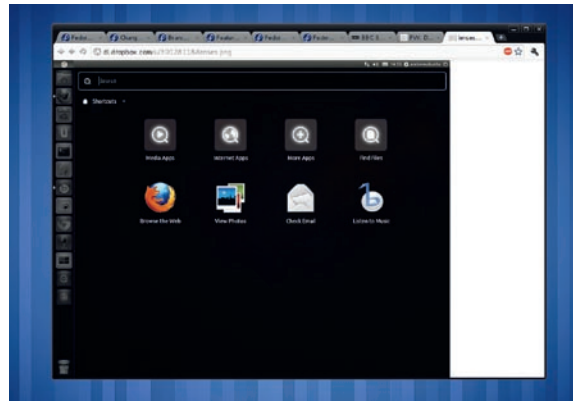
Галереи изображений

Кроме общей папки, вы увидите там папку **Pictures**. Если разместить в этой папке изображения на компьютере А и заглянуть в нее с компьютера В, файлы будут там, как мы видели

раньше. Однако если зайти на www.dropbox.com/photos с логином и паролем, то эти файлы отобразятся как онлайн-галерея. Более того, вас снабдят ссылкой, которую можно послать друзьям, не имеющим учетной записи на Dropbox. Это идеально, если вам приходится делиться фотографиями с личностями, страдающими крайней формой паранойи и бледнеющими при намеке на отправку их данных в онлайн-сервис, и дает шанс доставить фотографии родственникам, которые не терпят, когда к ним «суются», и по данной причине не намерены вступать в вашу мафию на Facebook.



» Чтобы поделиться файлами с тем, кто не зарегистрирован в Dropbox, щелкните правой кнопкой мыши по файлу в папке **Public** и выберите **Dropbox > Copy Public Link...**



» ...вставьте эту ссылку в электронное письмо, отошлите нужным получателям, и они смогут открыть этот файл в браузере.

OwnCloud/PackageCloud

Пользователи KDE, наверно, вознегодовали при упоминании *Nautilus*, как будто нас не волнует, что бывают и другие рабочие столы. Но мы тут ни при чем: так решили парни из Dropbox, написавшие клиент только для файлового менеджера Gnome. И это немного сурово, ведь то, что Dropbox для KDE не родной, заметно только при попытке открыть каталоги из папки Dropbox: они откроются в *Nautilus*, а не в *Konqueror* или *Dolphin*. Однако главная идея Dropbox – в гладкости его использования, и пользователи KDE достойны решения, не уступающего основному.

Решение, о котором идет речь – OwnCloud: эта система позволяет сохранять в облаке не только файлы, но и установки рабочего стола и приложений в KDE. Хотя она доступна для всех пользователей Linux, изначальное решение предложено на Camp KDE 2010, и было собрано для удобной интеграции с KDE. В обычной для KDE манере, пользователи KDE получают выгоду от сервиса большей сложности, но и с большими возможностями.

Хостить или не хостить

Использовать OwnCloud можно двумя способами: либо кто-нибудь установит его для вас на сайте хостинга PackageCloud (подробнее чуть ниже), либо можно установить его на своем компьютере и стать самому себе хостом платформы для облака. Преимущество собственной установки – все ваши данные у вас под контролем, и нет ограничений на объем хранимого в облаке, кроме тех, что накладывает ваше оборудование.

Имеется пакет для свежей версии Ubuntu, однако установив его, мы все еще получали сообщения об отсутствующих зависимостях, так что возьмите лучше версию для вашего дистрибутива с www.owncloud.org и обеспечьте все зависимости сами.

К счастью, их не так уж много: это Apache, PHP, MySQL и пара библиотек.

Гораздо более простое решение – создать учетную запись хостинга на www.packagecloud.com. Вам потребуется задать адрес электронной почты, имя пользователя и пароль; единственным препятствием здесь будет то, что **.co.uk** не распознается как правильный домен для электронной почты. Перейдите по ссылке, которую пришлют на ваш электронный ящик, и вы сможете войти в PackageCloud. Перейдите на вкладку APS Installer, отыщите пакет OwnCloud, нажмите на установку, и пакет установится в вашем облаке.

(Это кажется слишком сладким для этого мира, но дело в том, что размер хранилища очень мал – всего 500 МБ. За большие объемы придется платить.)

Вам пришлют два письма для подтверждения; после второго вы сможете зайти в учетную запись PackageCloud. Теперь зайдите в браузере на вкладку APS Installer и поищите «owncloud». Если вы решите установить его, вас попросят предоставить информацию о его расположении внутри вашего хранилища PackageCloud – у нас это <http://andrew1.packagecloud.com/owncloud>, а по окончании процесса установки вы сможете перейти по этому адресу прямо из браузера. Задайте пароль для созданной базы данных MySQL, имя пользователя-администратора, пароль, примите условия лицензии, нажмите на Install и подождите, пока пакет произведет все нужные действия. Зайдя в My Packages, вы увидите, что OwnCloud успешно установлен; теперь щелкните по ссылке Main Site и войдите в учетную запись OwnCloud. Начальный экран поначалу покажется неинтуитивным, но в More Actions вы увидите запросы для создания и загрузки файлов и папок. LXF

» Командная строка Оказывается, что не так она страшна, как ее малюют

КОМАНДЫ: GNU/

Тихон Тарнавский рассказывает, как средствами командной строки облегчить работу с Интернет-ресурсами.



Наш эксперт

Тихон Тарнавский работает в Linux-консоли больше восьми лет и точно знает, как сделать эту работу удобной.

В предыдущих статьях мы работали только с локальными данными. А на дворе давно век Интернета! Давайте теперь решим пару задач, связанных с сетью. Вполне возможно, у вас есть несколько (или даже много) «любимых» сайтов, которые вы регулярно проверяете на обновления. Многие сайты сейчас предоставляют подписки по RSS или электронной почте, но, к сожалению, не все. Более того, нет таких подписок, как правило, на сайтах или разделах сайтов, которые обновляются не слишком часто – а открывать каждый день одну и ту же неизменную страницу со временем становится скучновато, и мы все легче и легче об этом забываем. Так вспомним же известный принцип: «если ты не хочешь читать вывод программы, заставь это делать другую программу». Ведь для человека разницы между программой и сайтом в этом смысле нет, даже если на деле содержимое сайта не генерируется программой, а пишется вручную другим человеком (что, впрочем, последнее время бывает не так уж часто).

Первый пошел

Следуя еще одному принципу – «делай большое дело малыми частями» – для начала проследим за обновлением одного сайта. Уже когда все будет готово, повторим полученные действия для всех страниц из желаемого списка. Первое, что нам нужно сделать – загрузить нужную страницу. Поскольку эту страницу нужно будет с чем-то сравнить, лучше всего сохранить ее в файл. Для загрузки страниц из сети воспользуемся программой **wget**. Нам понадобится всего одна ее опция, которая позво-

ляет задавать имя файла сохраняемой страницы, '-O' [от слова output – вывод]:

```
wget linuxformat.ru/event -O ~/tmp/linuxformat.html
```

Двигаемся дальше. Нам нужно сначала сохранить текущую версию страницы, а затем точно таким же способом загружать новую и сравнивать с предыдущей. К тому же запуск такого сравнения лучше всего будет автоматизировать, а не вызывать его каждый раз вручную – ведь о ручном запуске ничуть не сложнее забыть, чем об открытии страницы в браузере. Поэтому лучше будет сразу начать записывать нужные команды в файл. Назовем этот файл **site-update** и сохраним в уже знакомом нам каталоге **~/bin/**.

Нам понадобится каждый раз сравнивать две версии одной и той же страницы. Исходя из этого, предлагаю версию, лежащую с прошлого раза, назвать «**имя-сайта.html**», а новую (которая останется лишь до окончания работы скрипта) – «**имя-сайта.tmp**». Сокращение «tmp», напомним на всякий случай, происходит от слова «temporary», то есть «временный». Поскольку мы хотим одну и ту же последовательность команд запускать и в первый раз, и при повседневной обработке, и, скажем, при добавлении новых страниц в список слежения, лучше всего будет исходить из того, что мы не знаем, есть ли у нас уже ее предыдущая сохраненная версия. Значит, изначально мы в любом случае сохраняем страницу во «временный» файл:

```
#!/bin/bash
```

```
wget linuxformat.ru/event -O ~/tmp/linuxformat.tmp
```

Теперь логично будет первым обработать тот случай, когда старой версии еще нет. Проверить ее наличие можно уже известной нам командой **test** или **[**. У нее есть множество различных ключей, позволяющих проверять, чем является (или не является) заданный параметр. Конечно, в понятие «не является (чем-то конкретным)» включается и вариант «не существует в принципе». Большинство опций команды **[** (как и многих других команд) – «говорящие». Так, опции **f** соответствует проверка на обычный файл **[file]**:

```
[ -f ~/tmp/linuxformat.html ]
```

Если такого файла не существует, нам нужно только создать его (точнее, переименовать уже сохраненную в предыдущей строке версию) и завершить работу – поскольку все дальнейшие проверки в этом случае бессмысленны. Можно было бы реализовать это оператором «**if... else... fi**». Но тогда ту часть, что мы пишем сей-

час, пришлось бы написать в конце (после **else**), а позже вставлять команды в середину. Это нарушает последовательность решения задачи, и мы так делать не будем. Как из любой безвыходной ситуации, тут существует минимум два выхода. Первый – записать «обратное» условие через оператор отрицания, обозначаемого восклицательным знаком (такое обозначение тоже будет привычно знакомым с азами программирования на некоторых языках):

```
[ ! -f ~/tmp/linuxformat.html ]
```

Но так как после переименования файла работа должна завершиться, то совершенно незачем запикивать все последующие

Linux и смекалка

команды внутрь “else... fi”. Вместо этого мы можем воспользоваться антиподом уже известного нам оператора ‘&&’ – ‘||’. Он предписывает выполнять следующую команду только в том случае, если предыдущая завершилась с ошибкой (то есть, в данном случае, заданное условие не выполнилось).

```
[ -f ~/tmp/linuxformat.html ] || нужные-команды
```

Однако мы умеем записывать после && или || только одну команду, а сейчас нам нужно несколько – переименование файла и завершение работы. Для этого в командной оболочке применяются также довольно широко распространенные в этих целях фигурные скобки:

```
{ команда1
  команда2
}
```

Или, в одну строку:

```
{ команда1; команда2; }
```

Обратите внимание: закрывающая скобка пишется обязательно с новой строки – то есть после разделителя (которым вместо перевода строки может выступать точка с запятой).

Перейдем к «внутренностям» скобок. В первую очередь нам нужно переименовать файл. Мы такой команды пока не знаем, но легко можем ее найти с помощью **man -k rename** [rename – переименовать]:

```
$ man -k rename
```

```
[...]
```

```
mv (1) - move (rename) files
```

Вызывается она в нашем случае более чем просто: “**mv старое-имя новое-имя**”. А команда завершения работы называется столь очевидно, что ее даже искать не придется – **exit** [выход]. Итак:

```
#!/bin/bash
```

```
wget linuxformat.ru/event -O ~/tmp/linuxformat.tmp
```

```
[ -f ~/tmp/linuxformat.html ] || {
```

```
  mv ~/tmp/linuxformat.tmp ~/tmp/linuxformat.html
```

```
  exit
```

```
}
```

Теперь у нас наверняка есть две версии страницы, можем переходить к сравнению. Вывод команды **man -k compare** [compare – сравнивать] может оказаться довольно большим. Чтобы легче было искать, вспоминаем, что сравнивать нам требуется файлы, и добавляем фильтр: **man -k compare | grep files**. Теперь нетрудно найти нужное:

```
diff (1) - compare files line by line
```

т. е. «сравнить файлы построчно». По умолчанию эта команда выводит различия между текстовыми файлами. Нам различия не нужны, а нужно лишь проверить сам факт полного совпадения файлов. Для этого предназначена опция **q**.

Давайте проверим ее на одинаковых и на разных файлах. Но чтобы не создавать файлы специально для этой проверки, воспользуемся еще одной конструкцией подстановки: ‘<(…)’. Она создает временный псевдофайл в файловой системе устройств

```
$ diff -q <(echo 1) <(echo 1) && echo Ok
Ok
$ diff -q <(echo 1) <(echo 2) && echo Ok
Файлы /dev/fd/63 и /dev/fd/62 различаются
$
```

/dev/ и подставляет его имя в заданном месте. С опцией **q** команда **diff**, как и **grep**, завершается корректно либо с ошибкой в зависимости от результата, поэтому код ее завершения мы тоже проверим (см. рис. 1). Как видно из этой проверки, «безошибочным» считается случай одинаковых файлов. А вот в случае разных файлов кроме «ошибочного» результата выводится текстовое сообщение, которое нам при автоматической обработке совершенно не нужно. Для таких случаев в Unix-системах существует специальное псевдоустройство в той же файловой системе **/dev:/dev/null**. Достаточно перенаправить вывод в него, чтобы тот, образно говоря, исчез. Значит, целиком нужная нам команда будет выглядеть так:

```
diff -q ~/tmp/linuxformat.html ~/tmp/linuxformat.tmp >/dev/null || что-то делаем
```

А что будем делать, если содержимое файла изменилось? Видимо, стоит предложить открыть этот сайт в браузере; но и дать возможность отказаться от этого. Значит, здесь подходит известная нам с прошлого раза программа **zenity**. Например, так:

```
diff -q ~/tmp/linuxformat.html ~/tmp/linuxformat.tmp >/dev/null || zenity --question --title="Сайт обновился" --text="Сайт linuxformat.ru/event обновился. Хотите открыть его в браузере?"
```

При положительном ответе нужно запустить браузер с заданным именем сайта (если браузер уже открыт, сайт, скорее всего, откроется в его новой вкладке или новом окне). Операторы **||** и **&&** можно составлять в цепочки любой длины. Но в цепочке “**команда1 || команда2 && команда3**” третья команда выполнится при удачном завершении хотя бы одной из первых двух. Как быть? Вариантов два. Можно вместо “**команда2 && команда3**” воспользоваться оператором **if**: “**if команда2; then команда3; fi**”. А можно заключить последние две команды в фигурные скобки – на этот раз не для их объединения между собой (они и так объединены оператором **&&**), а для отделения от первой. Последний штрих: если браузер еще не открыт, он запустится как отдельный процесс, и хорошо бы, чтобы скрипт не ждал его закрытия, а завершился сразу. Для этого можно запустить нужную команду в фоновом

► Рис. 1. Сравнение файлов командой **diff**.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

```
#!/bin/bash
dir=~/.site-update
mkdir -p $dir
for site in $(<$dir/site-list); do
  file=$dir/${site//\//+}
  wget $site -O $file.tmp
  [ -f $file.html ] || {
    mv $file.tmp $file.html
    continue
  }
  diff -q $file.html $file.tmp || {
    zenity --question --title="Сайт обновился" \
      --text="Сайт $site обновился. Хотите открыть его в браузере?" &&
    firefox $site &
  }
  mv $file.tmp $file.html
done &>/dev/null
```

➤ Рис. 2. Скрипт, проверяющий, не обновилась ли сайты из списка.

режиме, добавив после нее одиночный символ **&**. После чего заменим старую версию файла новой, чтобы в следующий раз сравнивать уже с ней. Итак:

```
#!/bin/bash
wget linuxformat.ru/event -O ~/tmp/linuxformat.tmp
[ -f ~/tmp/linuxformat.html ] || {
  mv ~/tmp/linuxformat.tmp ~/tmp/linuxformat.html
  exit
}
diff -q ~/tmp/linuxformat.html ~/tmp/linuxformat.tmp >/dev/null ||
zenity --question --title="Сайт обновился" --text="Сайт linuxformat.ru/event обновился. Хотите открыть его в браузере?" &&
firefox linuxformat.ru/event &
mv ~/tmp/linuxformat.tmp ~/tmp/linuxformat.html
```

Огласите, пожалуйста, весь список

Теперь переходим к отслеживанию нескольких сайтов. Во-первых, раз сохраненных страниц будет несколько, лучше не захламлять ими каталог `~/tmp/`, а создать свой. Традиционно таким «личным каталогам программ» дают имена, начинающиеся с точки, и размещают их в домашнем каталоге. Так что назовем его `~/site-update`:

```
mkdir -p ~/.site-update
Команда mkdir – сокращение от “make directory” [создать каталог], а ключ p делает две вещи: во-первых, все родительские каталоги, если их еще нет, тоже будут созданы; а во-вторых, не выводится сообщение об ошибке, если заданный каталог уже есть. Это избавляет нас от дополнительной проверки: мы можем не беспокоиться, что нужный каталог может уже существовать. Но поскольку этот каталог будет и дальше использоваться в скрипте, лучше его сохранить в переменной, а затем эту переменную подставлять. Переменные shell (командной оболочки) назначаются конструкцией имя=значение (обязательно без пробелов вокруг знака равенства), а подстановка значений выглядит как $имя.
```

```
dir=~/.site-update
mkdir -p $dir
Список адресов лучше хранить в отдельном файле, а не в самом скрипте – так его удобнее будет редактировать впоследствии, да и обрабатывать тоже. Файл этот резонно держать в том же каталоге, а назвать его можно site-list [список сайтов]. Для повторения всех уже записанных нами действий для каждого сайта из списка воспользуемся уже известным нам циклом for. Но в нем элементы списка нужно указывать прямо в самой команде, а не получать из файла. Для этого совместим еще две знакомые конструкции: подстановку результатов выполнения – $(...) или `...` – и ввод из файла – <.
```

```
for site in $(<$dir/site-list); do
```

Теперь нужно подставить сюда все написанное раньше. Тут возникает вопрос, как быть с именами файлов. Адреса страниц в неизменном виде использовать в качестве имен нельзя, так как в них могут быть слэши, а в самих именах слэши недопустимы, ведь они используются как разделители каталогов. Самое простое решение – заменить слэш другим символом, который вряд ли будет встречаться в адресах страниц на том же месте, что и слэш. Например, знаком плюса. Для столь простой замены даже не нужно прибегать к специальным утилитам – достаточно встроенного функционала **bash**. Имя файла сейчас хранится в переменной **site**, а для замены внутри переменных существуют такие конструкции: **\$(имя/было/будет)** и **{имя//было/будет}**. Вариант с одиночным первым слэшем заменяет только первое вхождение заданного текста, а со двоек – все. Значит, нам нужен второй вариант. Раз заменять нам нужно тот же слэш, который используется здесь как «управляющий» символ, его придется «экранировать» обратным слэшем. Конструкция получится не слишком удобочитаемая, но главное, что она будет работать. Поскольку сам неизменный адрес нам тоже еще пригодится, сохраним результат в другой переменной, оставив первую нетронутой. А раз сохраняем мы имя файла, то добавим к нему сразу и имя каталога, чтобы потом не писать его каждый раз:

```
file=$dir/${site//\//+}
```

Наконец, редактируем все, что у нас уже было, подставляя нужное имя файла, и завершаем цикл. И делаем пару последних штрихов. Теперь при отсутствии одного из файлов нам нужно прервать не весь скрипт, а только один проход цикла, переходя к следующему. Это делается командой **continue** [продолжить], на которую мы и заменим **exit**. Второе: мы «заземлили» вывод команды **diff**, но команда **wget** тоже выводит сообщения о загрузке файлов. Причем здесь такое же перенаправление не поможет, так как выводит она их не на стандартный вывод (stdout), а в поток стандартных ошибок (stderr). По умолчанию оба потока направлены на управляющий терминал, но управлять ими можно (и нужно) по отдельности – для этого они и разделены. Перенаправление стандартных ошибок делается парой символов **2>** (без пробела) вместо **>**. Но чем добавлять второе перенаправление в `/dev/null`, мы можем поступить гораздо хитрее. Нам не нужен вообще никакой вывод от этого скрипта, ведь все общение с человеком идет через диалоговое окно **zenity**.

И тут очень кстати оказывается еще одно полезное свойство **bash**: вся конструкция **for... do... done** (да и не только она, а и **if... else... fi**, и другие аналогичные) в смысле ввода-вывода воспринимается как одна команда. В частности, можно поставить значок перенаправления после **done**, «заземлив» весь цикл сразу. А кроме того, можно перенаправить в одно и то же место сразу оба выходных потока – стандартного вывода и ошибок – парой символов **&>** (тоже без пробела). Теперь можем любоваться результатом (рис. 2).

Четко по расписанию

Осталась завершающая часть: запускать полученный скрипт автоматически с заданной периодичностью. Запуском таких периодичных заданий занимается системный демон **cron**. Да, он умеет запускать не только системные задания, но и пользовательские. Для этого существуют специальные пользовательские файлы **crontab** [cron table], управлять которыми можно с помощью одноименной программы. Программа это весьма проста в использовании и имеет всего три основных опции: **e** [edit] – редактировать файл **crontab**; **l** [list] – вывести текущее содержимое этого файла; и **r** [remove] – удалить файл. Справку по синтаксису файла **crontab** можно получить командой **man 5 crontab**. Если вы в свое время

➤ Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

ознакомьтесь с man-страницей самой команды **man**, то из секции «смотри также» в **man crontab** вам должно быть это понятно. Кроме того, при первом запуске **crontab -e** созданный файл скорее всего будет щедро откомментирован (см. рис. 3).

Как видите, каждое задание нужно записывать одной строкой, состоящей из шести полей, разделенных пробелами. В первых пяти полях задается периодичность задания, а в шестом – само задание (команда). Пробелы внутри шестого поля не учитываются, то есть можно прямо там записывать команды вместе с аргументами. Первые пять полей имеют такие значения:

- 1 минута
- 2 час
- 3 день месяца
- 4 месяц
- 5 день недели

В каждом из полей может стоять либо звездочка, либо число или перечень чисел. Звездочка означает «любой». То есть если поставить звездочки во всех пяти полях, задача будет выполняться ежеминутно без перерывов. Перечень чисел можно задать в явном виде, перечислив через запятую отдельные числа или их диапазоны: например, если в пятом поле написать “1,2,4-7”, то задание будет выполняться во все дни, кроме среды. А можно задать и неявно, в виде “*/число”. Если * означает «каждый» (день, месяц и т.д.), то, скажем, */2 – каждый второй, а */5 – каждый пятый.

Как видите, настраивать периодичность можно очень гибко. Например, так можно запланировать задание на полдень первой субботы каждого месяца: “0 52 5-7 * 6”. Или, в качестве шутки: если вы суеверны и боитесь пятницы тринадцатого, можете поставить какое-нибудь зловещее уведомление на полночь таких дней, в каком бы месяце они ни выпали – “0 5 13 * 5”.

Кстати о полночи. Задания выполняются cron’ом только четко в заданное время. Если компьютер в это время был выключен, задание не выполнится совсем. Домашние компьютеры редко держат включенными круглосуточно и круглый год. Поэтому, если вы найдете удобным назначать компьютеру автоматические задания, которые не захотите пропускать из-за его выключения, можете использовать для этого пакет *anacron* [anachronistic cron], который запускает и пропущенные задания – при первом включении компьютера после запланированного по расписанию времени. У файлов *anacrontab* синтаксис еще проще (хотя, казалось бы, куда уж проще?), так что оставляю его вам для самостоятельного изучения.

Но вернемся к нашим web-сайтам. Периодичность проверок выбирайте на свой вкус, в зависимости от того, как часто обновляются выбранные сайты и насколько интенсивно вы собираетесь за ними следить. Например, каждые десять минут:

```
*/10 * * * * ~/bin/site-update
```

Или – по пятницам после обеда:

```
15 14 * * 5 ~/bin/site-update
```

Но тут возникает еще одна задача. Мы-то при проверке запускали скрипт из «иксов». И в этих же «иксах» у нас отображалось окно **zenity** и вызывался браузер. А задания **cron** об «иксах» ничего не знают. Так что и скрипт, запускаемый с помощью этих заданий, знать о них ничего не будет. Значит, нужно ему об этом сказать. Для этого есть стандартный для всех Unix-подобных систем механизм общения между процессами (работающими программами), называемый переменными окружения. Среди этих переменных есть «системные», со специальными зарезервированными именами; такие имена, как правило, пишутся заглавными буквами. Каждая переменная создается неким одним процессом и доступна только его потомкам, то есть процессам, вызванным из него или других его потомков. Нужная нам переменная «принадлежит» X-серверу и хранит его «имя», а называется она **DISPLAY**. Соответственно, для того, чтобы запустить

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
-UU-:@:---F29 crontab All (1,0) (Text ll Fly Server)-----
```

Рис. 3. Исходный вид пользовательского файла **crontab**.

программу на уже работающем X-сервере, нужно передать ей правильное значение переменной **DISPLAY**. Посмотреть нужное значение можно из «иксового» терминала привычной командой **echo**, поскольку переменные окружения и переменные командной оболочки – это по своей сути одно и то же (с одной оговоркой, о которой чуть ниже):

```
$ echo $DISPLAY
:0.0
```

Значит, нам нужно задать этой переменной подходящее значение прямо в скрипте. Для этого обычно сначала задаются переменные оболочки, которые затем экспортируются в то самое «окружение» с помощью команды **export**:

```
переменная1=значение1
переменная2=значение2
export переменная1 переменная2
```

Но если переменная одна, экспорт и назначение можно совместить и написать вот так:

```
export DISPLAY=:0.0
```

Добавляйте эту строчку в начальную часть скрипта (рис. 4), составляйте список сайтов, создавайте задание **cron** – и ваш очередной личный помощник готов. **LXF**

```
#!/bin/bash
dir=~/site-update
mkdir -p $dir
export DISPLAY=:0.0
for site in $(<$dir/site-list); do
  file=$dir/${site//\//+}
  wget $site -O $file.tmp
  [ -f $file.html ] || {
    mv $file.tmp $file.html
    continue
  }
  diff -q $file.html $file.tmp || {
    zenity --question --title="Сайт обновился" \
      --text="Сайт $site обновился. Хотите открыть его в браузере?" &&
    firefox $site &
  }
  mv $file.tmp $file.html
done &>/dev/null
-UUU-:@:---F17 site-update All (19,0) (Shell-script[bash] Server)-----
```

Рис. 4. Окончательный вариант скрипта, который можно запускать с помощью **cron**.

Erlang: Коллекции

Андрей Ушаков принимает за новую базовую сущность: говорит – коллекции, подразумевает – списки. Но не только списки.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

Мы продолжаем обзор базовых сущностей языка Erlang. Пришла пора поговорить о коллекциях. Когда мы говорим о коллекциях, то в первую очередь вспоминаем о списках. И действительно, списки – это наиболее часто используемая коллекция. Более того, списки часто являются базой для других коллекций и сущностей. Так, в языке Erlang списки являются базой для работы со строками (с не Unicode-строками; для Unicode-строк базой являются двоичные данные). Но разговор о строках будет в одном из следующих номеров.

Что же такое списки и для чего они нужны? Список – это контейнер для данных, обработка которых будет происходить одним и тем же способом. При этом количество данных (элементов списка) может быть любым. Не обязательно, чтобы тип всех элементов списка был одним и тем же: вполне возможна ситуация, когда список содержит данные различных типов. Главное, что мы всегда предполагаем, что все элементы списка будут обработаны одинаковым (или похожим) образом. Таким образом, списки нам нужны для хранения и обработки переменного количества похожих (с точки зрения обработки) данных. Каждый элемент списка располагается в определенной позиции, и получается, что элементы в списке упорядочены в соответствии с занимаемой ими позицией. Если мы добавляем элементы строго в конец списка, то в таком случае элементы списка будут упорядочены в соответствии с очередностью их добавления (в случае, например, множества это будет не так). Списки очень похожи на векторы из стандартной библиотеки C++, за одним исключением: списки в Erlang неизменяемые (как и все другие объекты).

Как объявить переменную типа список? Как мы помним (LXF143), Erlang – это язык со строгой динамической типизацией. Это означает, что тип переменной определяется в момент ее инициализации. Инициализатор для списка выглядит очень просто: внутри квадратных скобок “[” и “]” мы через запятую перечисляем значения элементов списка. Например, мы объявляем переменную типа список с ее инициализацией следующим образом: `ListVar = [12, "34", 56]`. Количество элементов в списке – это его размер. Минимальный размер списка – 0 элементов (пустой список). Максимальный размер списка ничем не ограничен (если быть точным, то максимальный размер списка ограничен только наличием – или отсутствием – свободной памяти).

Сразу же возникает вопрос: как получить доступ к отдельным элементам списка? Есть две возможности: использовать функцию `lists:nth/2` (о ней мы поговорим ниже) или операцию соответствия шаблону [pattern matching]. Операция соответствия шаблону (в отличие от кортежей) не позволяет получить доступ к произвольному элементу списка: она позволяет всего лишь обратиться к головному элементу и оставшейся части списка. Выглядит это следующим образом: `[Head | Tail] = [1, 2, 3, 4]`. Здесь переменная **Head** получит значение 1, а переменная **Tail** – [2, 3, 4]. Очевидно, что в качестве головного в шаблоне можно задать несколько переменных; эти переменные будут сопоставляться (а если у них нет значения, то получать соответствующие значения) первым нескольким (по числу переменных в шаблоне) элементам списка. И вполне логично, что если мы зададим в головном шаблоне переменных больше, чем элементов в списке, то такой список операцию соответствия шаблону не пройдет. Так, например, следующая операция соответствия шаблону выполняется успешно: `[H1, H2 | Tail] = [1, 2, 3]` и переменные **H1**, **H2**, **Tail** получают значения 1, 2 и [3] соответственно. А следующие две операции соответствия шаблону не выполняются: `[1, H2 | Tail] = [2, 3, 4]` и `[H1, H2, H3 | Tail] = [1, 2]`. Подобный (и кажущийся на первый взгляд ограниченным) вид операции соответствия шаблону позволяет рекурсивно обрабатывать весь список поэлементно. Например, функцию нахождения суммы всех элементов в списке можно написать следующим образом:

```
list_sum(NumberList) -> list_sum_impl(NumberList, 0).
list_sum_impl([], Sum) -> Sum;
list_sum_impl([Number | Rest], Sum) -> list_sum_impl(Rest, Sum + Number).
```

За счет операции соответствия шаблону мы рекурсивно поочередно обрабатываем (суммируем) значения элементов, а благодаря хвостовой рекурсии (см. LXF145), подсчет суммы элементов списка разворачивается в цикл.

Помимо операции соответствия шаблону, для списков (списочных типов данных) определено еще две операции: “++” и “--”. Операция “++” – это операция сложения двух списков: она создает и возвращает новый список (мы помним, что все объекты в языке Erlang неизменяемые), в котором сначала идут все элементы из первого списка, а потом все элементы из второго. Операция “--” более сложна: она возвращает новый список, который содержит

КАК СУЩНОСТИ

элементы, составляющие разность содержимого первого и второго списков. Алгоритм построения этой разности следующий: сначала копируются все элементы из первого списка, а потом для каждого элемента из второго списка ищется первое вхождение этого элемента в копию; если такое вхождение будет найдено, то из копии данный элемент удаляется, и дальнейшая работа продолжается уже с измененной копией. Об этом будет сказано ниже, но операция разности двух списков очень тяжелая: ее сложность составляет $O(M \cdot N)$, где M – размер первого списка, N – размер второго.

Как мы видели выше, для создания списков можно использовать инициализаторы и операции сложения и разности. Но в общем случае это далеко не всегда удобно, хотя, используя эти операции и рекурсивные функции, можно создать любой список. Поэтому в языке Erlang существует техника, легко позволяющая создавать списки в соответствии с очень сложными правилами: эта техника называется конструирование списков [List Comprehensions]. Имеются две сущности, появляющиеся в выражении конструирования списков: генераторы и фильтры.

Генератор – это сущность, порождающая некоторую последовательность. В качестве генератора может выступать любое выражение, результатом выполнения которого является список. Чтобы иметь доступ к элементам генерируемого списка, с каждым генератором связывают переменную с уникальным (в пределах данного выражения конструирования) именем. Таким образом, выражение для генератора имеет следующий вид: **Pattern** <- **ListExpr**, где **Pattern** – шаблон для операции соответствия, **ListExpr** – выражение, результатом выполнения которого является список. Шаблоном **Pattern** обычно является имя переменной, используемой для доступа к элементам генерируемого списка; однако, если выражение **ListExpr** возвращает список «сложных» объектов (например, кортежей), то шаблон может быть более сложным.

Фильтр – это выражение, возвращающее **true** или **false**. Фильтр позволяет возвращать не все значения от генератора, а только те, которые удовлетворяют некому условию.

А теперь давайте соберем все вместе. Выражение конструирования списков имеет следующий вид: **[Expr | Qualifier1, ..., QualifierN]**, где **Qualifier1**, ..., **QualifierN** – это либо выражение генератора, либо выражение фильтра, а **Expr** – выражение для формирования элементов итогового списка,

в которое могут входить любые переменные из области видимости выражения **Expr** (в том числе и все переменные, связанные с генераторами). Следует также упомянуть,

что и в выражения фильтра также могут входить любые переменные, но уже из области видимости выражения фильтра: это означает, что в выражении фильтра могут использоваться только те переменные для генераторов, которые располагаются левее определения выражения фильтра. Как все это работает?

Вычисляются все возможные комбинации элементов из выражений генераторов (в том порядке, в котором эти выражения генераторы объявлены), после чего эти комбинации пропускаются

через выражения фильтра. Все те комбинации, для которых все выражения фильтра вернули **true**, формируют элементы итогового списка при помощи вычисления выражения **Expr** для каждой такой комбинации. С первого взгляда кажется, что все это очень сложно, но на самом деле это не так. Давайте рассмотрим несколько примеров, чтобы концепции конструирования списков улеглись в голове. Для начала возьмем пример простого комбинирования элементов от двух генераторов без фильтров (так называемое Декартово произведение):

```
[{X, Y} || X <- [1, 2], Y <- ["a", "b"]].
```

Результат вычисления этого выражения очевиден – это будет список

```
[{1, "a"}, {1, "b"}, {2, "a"}, {2, "b"}].
```

В следующем примере мы будем возвращать список пар чисел, чья сумма четная:

```
[{X, Y} || X <- [1, 2, 3], Y <- [1, 2, 3], (X+Y) rem 2 == 0].
```

Очевидно, что результатом будет следующий список:

```
[{1, 1}, {1, 3}, {2, 2}, {3, 1}, {3, 3}].
```

А теперь приведем примеры сложнее. Первый пример – это реализация алгоритма быстрой сортировки [quick sort]:

```
sort([Middle | T]) -> sort([X || X <- T, X < Middle]) ++ [Middle] ++ sort([X || X <- T, X >= Middle]);
sort([]) -> [].
```

Второй пример – это реализация методов фильтрации [filter] и преобразования [map] списков:

```
map(Fun, List) -> [Fun(X) || X <- List].
filter(Pred, List) -> [X || X <- List, Pred(X)].
```

И, наконец, следует заметить следующее: в выражениях конструирования списков помимо генераторов списочных данных могут появляться и генераторы битовых данных (битовых строк), результат которых преобразуется в списочные данные. Но разговор о битовых данных (битовых строках) – это тема одной из следующих статей.

Выше мы показали в качестве примеров реализацию методов сортировки, фильтрации и преобразования списков. В реальных программах так делать не стоит: все эти и большое количество других функций уже реализованы в стандартных (поставляемых вместе с компилятором и средой выполнения) библиотеках языка Erlang. Часть этих функций является BIF. В первую очередь из этих

BIF следует упомянуть следующие: **hd/1** возвращает головной элемент списка, **tl/1** возвращает остаток списка без головного элемента, **length/1** возвращает количество элементов списка. Помимо них, есть еще

следующие BIF: **is_list/1** позволяет определить, является ли тип выражения списком, а набор функций **list_to_xxx/1** позволяет преобразовать список во что-то еще – например, в кортеж. Следует упомянуть, что следующие BIF можно применять в выражениях охраны (см. **LXF143**): **hd/1**, **tl/1**, **length/1**, **is_list/1**. Также существует достаточно большой набор функций (определенных в модуле **lists**) для более сложных задач работы со списками, включая фильтрацию, преобразование, вычисление агрегатов и многое другое. »

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Давайте рассмотрим некоторые, как мне кажется, наиболее полезные функции из этого модуля. Функция `lists:filter(Pred, SourceList)` фильтрует исходный список `SourceList`: на выходе мы будем иметь список, содержащий только те элементы исходного списка, для которых предикат `Pred/1` вернет значение `true`. Следующий пример возвратит список только четных чисел:

```
lists:filter(fun(Number) -> Number rem 2 == 0 end, [1, 2, 3, 4]).
```

Функция `lists:map(TransformFun, SourceList)` преобразует исходный список `SourceList` поэлементно: каждый элемент исходного списка преобразуется при помощи функции `TransformFunc/1`. Так, следующий пример возвращает список квадратов чисел из исходного списка:

```
lists:map(fun(Number) -> Number*Number end, [1, 2, 3, 4]).
```

Функция `lists:foldl(FoldFun, Acc0, SourceList)` позволяет вычислить агрегат по исходному списку `SourceList`. Вычисление происходит следующим способом: начиная с начального значения агрегата `Acc0`; элементы списка обходятся слева направо (в прямом порядке) и для каждого элемента и предыдущего значения агрегата вычисляется новое значение агрегата (при помощи функции `FoldFun(Element, PrevAcc)`); значение агрегата, вычисленное для последнего элемента, становится результатом, возвращаемым функцией `lists:foldl/3`. Существует вариант функции `lists:foldl/3`, в котором обход элементов при вычислении агрегата осуществляется не слева направо, а справа налево (в обратном порядке) –

`lists:foldr/3`. Ее наличие оправдывается ситуациями, когда значение получаемого агрегата может различаться в зависимости от направления обхода элементов (когда операция построения агрегата некоммукативна). Например, такая ситуация возникает, если мы хотим построить произведение матриц, расположенных в списке (операция произведения матриц, как известно, некоммукативна). Попробуем вычислить произведение чисел, заданных в исходном списке:

```
lists:foldl(fun(Number, Product) -> Number*Product end, 1, [1, 2, 3, 4]).
```

Следует отметить, что для вычисления суммы элементов списка использовать функцию `lists:foldl/3` не обязательно: для этой операции специально введена функция `lists:sum/1`. Следующие две функции позволяют проверить список в целом на выполнение некоторого условия: `lists:all(Pred, SourceList)` и `lists:any(Pred, SourceList)`. Функция `lists:all/2` возвращает `true`, если для всех элементов исходного списка выполняется условие, заданное функцией-предикатом `Pred`; функция `lists:any/2` возвращает `true`, если хотя бы для одного элемента исходного списка выполняется условие, заданное функцией-предикатом `Pred`. Вот как можно проверить, что все числа в списке больше 0:

```
lists:all(fun(Number) -> Number > 0 end, [1, 2, 3, 4]).
```

или – что хотя бы одно число больше 0:

```
lists:any(fun(Number) -> Number > 0 end, [-1, 2, -3, 4]).
```

В модуле `lists` есть еще масса полезных функций. Например, функции для сортировки: `lists:sort/1`, `lists:sort/2`, функции для сортировки без дубликатов: `lists:usort/1`, `lists:usort/2`, функция для доступа к элементу списка по индексу `lists:nth/2` и т.д. Перечисление всех их заняло бы слишком много места, но желающие могут посмотреть описание всех функций из модуля `lists` в документации по языку Erlang.

А мы пойдем дальше. Помимо списков, есть еще другие виды коллекций. В первую очередь это множества и словари. Зачем же они нужны, если аналоги множеств и словарей можно реализовать при помощи списков? Затем, что набор API для работы

со списками не предназначен для работы со списками как со словарями и множествами. Особенно это касается операций добавления и удаления элементов: обычно множества и словари представляют собой либо деревья, либо хэш-таблицы, и добавление и удаление элементов должно производиться особым способом. Кроме того, временные характеристики операций в списках и во множествах и словарях отличаются. Так, например, операция поиска для списка имеет сложность $O(N)$ (для отсортированного списка операция двоичного поиска имеет сложность $O(\log N)$, но при этом возникают накладные расходы на сортировку и сложности добавления и удаления элементов в такой список). А для множеств и словарей операция поиска имеет сложность либо $O(\log N)$, либо $O(1)$, в зависимости от структуры данных, использованной при реализации. Поэтому использование списков в тех случаях, когда данные по своей природе представляют собой либо множество, либо словарь, не разумно (хотя и возможно в случае коллекций небольшого размера).

Рассмотрим поближе реализации этих коллекций (множеств и словарей) в библиотеке языка Erlang. Библиотека языка Erlang содержит две реализации множеств, причем их интерфейс одинаков (под интерфейсом понимается множество экспортируемых из модуля функций): в модулях `sets` и `ordsets`. Между этими реализациями есть два различия. Во-первых, элементы сравниваются

на равенство при помощи разных операторов: в реализации в модуле `ordsets` используется оператор `"=="`, а в реализации в модуле `sets` – оператор `"=:"`. Разница между этими операторами в том, что оператор `"=:"`

«Библиотека языка Erlang содержит две реализации множеств.»

сравнивает два операнда такими, какие они есть, а оператор `"=="` может преобразовать один операнд к другому, если они имеют разные типы. Так, например, `1 =:= 1.0` вернет `false`, т.к. операнды имеют разные типы, а `1 == 1.0` вернет `true`, т.к. целочисленный операнд 1 может быть преобразован в вещественный операнд 1.0. Во-вторых, реализация множеств в модуле `sets` не определена и может быть любой (на данный момент – с помощью хэш-таблицы), а множества в модуле `ordsets` реализованы с помощью упорядоченного списка. Точно так же, в языке Erlang есть две реализации словарей: в модуле `dict` и в модуле `orddict`. Реализация словарей в модуле `dict` не определена и может быть любой (на данный момент – с помощью хэш-таблицы); для сравнения элементов используется оператор `"=:"`. Словари в модуле `orddict` реализованы с помощью списка пар (кортежей) ключ-значение, упорядоченных по ключу; для сравнения элементов используется оператор `"=="`.

Давайте посмотрим на практике, как работать со словарями и множествами. В следующем примере мы сначала создаем множество, потом добавляем в него два элемента, потом удаляем отсутствующий во множестве элемент и в конце получаем список всех элементов множества. Заметьте, что каждая операция, изменяющая множество, на самом деле создает новый объект (т.к. все объекты в языке Erlang неизменяемые).

```
Set0 = ordsets:new().
```

```
Set1 = ordsets:add_element(abc, Set0).
```

```
Set2 = ordsets:add_element(bac, Set1).
```

```
Set3 = ordsets:del_element(ccc, Set2).
```

```
List0 = ordsets:to_list(Set3).
```

Теперь рассмотрим пример со словарями: мы так же, как в предыдущем примере, сначала создаем словарь, потом добавляем две пары ключ – значение, потом удаляем данные из словаря по отсутствующему ключу и в конце получаем список всех пар ключ – значение, хранящихся в словаре.


```
Dict0 = orddict:new().
Dict1 = orddict:append(ccc, ccc_333, Dict0).
Dict2 = orddict:append(aaa, aaa_111, Dict1).
Dict3 = orddict:erase(bbb, Dict2).
List0 = orddict:to_list(Dict3).
```

Как мы увидели, списки, пожалуй, наиболее универсальная коллекция: на ее базе можно реализовать все другие коллекции. Однако у списков в языке **Erlang** имеется один минус: нет простого способа установить значение элемента в определенной позиции списка (пусть даже с созданием нового) – хотя в таких языках программирования, как C, C++, Java, C#, списки и их аналоги (векторы, массивы и т.д.) легко позволяют нам сделать это. Тем не менее разработчикам не придется писать свою реализацию списка с возможностью установки значения элемента в произвольной позиции: в библиотеке языка **Erlang** подобная реализация существует. Она называется массив и находится в модуле **array**. В массивах, в отличие от списков, индексация элементов начинается с 0, а не с 1; кроме того, массив можно создать фиксированного, а не динамического размера. Ну и, конечно, массивы позволяют задать значение элемента в определенной позиции; правда, как и в случае других объектов в языке Erlang, при этом будет создан новый объект. Покажем это на примере – создадим массив фиксированного размера (с размером 10 и значением по умолчанию -1), потом установим значение первого элемента (по индексу 0) равным 100, после чего получим значение второго элемента (по индексу 1) и, наконец, получим представление массива в виде списка:

```
Array0 = array:new(10, [fixed, {default, -1}]).
Array1 = array:set(0, 100, Array0).
Value0 = array:get(1, Array1).
List0 = array:to_list(Array1).
```

Мы рассмотрели списки и другие коллекции: теперь пришло время поговорить об эффективности использования тех или иных операций над коллекциями. Начнем разговор со следующих операций над списками: “++” и “--”. Операция “++” реализована следующим образом: она создает новый список, копируя все элементы из левого операнда, но элементы из правого операнда не копируются, а вместо этого создается ссылка на правый операнд. Все объекты в языке Erlang неизменяемые, и можно быть уверенным,

что правый операнд будет всегда содержать один и тот же набор элементов. Поэтому, когда мы создаем новый список операций “++”, добавляя элементы один за другим, очень важно с точки зрения производительности, чтобы элементы добавлялись спереди (во избежание ненужных операций копирования). Что же делать, если нам требуется обратный порядок построения списка? Все очень просто: построим список в порядке наибольшей эффективности, после чего при помощи функции **lists:reverse/1** сменим порядок следования элементов на обратный.

Теперь займемся операцией “--”. Эта операция строит список, являющийся разностью левого и правого операнда. Выше был рассмотрен алгоритм построения разности; сложность этой операции составляет $O(M*N)$, где **M** – размер первого списка, **N** – размер второго. Очевидно, что в случае больших списков эта операция займет много времени, и ее следует избегать (как и других операций из теории множеств, например, пересечения двух множеств).

Сложность операций над коллекциями неодинакова. Скажем, сложность функции **lists:max/1** будет $O(N)$ (т.к. мы не располагаем информацией об упорядоченности списка), а сложность операции **lists:sort/1** – $O(N \log N)$.

Хочется отдельно остановиться на сложности доступа к элементам во множествах и словарях. У нас есть две реализации множеств (модули **sets** и **ordsets**) и словарей (модули **dict** и **orddict**). Реализации в модулях **ordsets** и **orddict** используют сортированный список, поэтому доступ к элементу (паре ключ–значение) будет иметь сложность $O(N \log N)$. Реализации в модулях **sets** и **dict** стандартом не определены, но на данный момент используются хэш-таблицы, поэтому доступ к элементу (к паре ключ–значение) будет иметь сложность $O(1)$. Однако тут есть одно «но»: вполне возможна ситуация, когда мы сохраняем в хэш-таблицу элементы, хэш-коды которых попадают в одну ячейку; в этом случае сложность доступа к элементу возрастает до $O(N)$.

В данной статье мы рассмотрели списки и другие коллекции. Но это не последняя статья про них: в одном из грядущих номеров статья будет про строки, базой которых являются списки; отдельную статью также планируется посвятить некоторым аспектам работы со строками, например, конструированию списков. А следующая статья будет посвящена такой интересной сущности языка Erlang, как битовые строки (двоичные данные). **LXF**

Полезные заметки

» O-нотация

O-нотация – математические обозначения для сравнения асимптотического поведения функций. Используются в различных разделах математики, но активнее всего – в математическом анализе, теории чисел и комбинаторике, а также при оценке сложности алгоритмов. В частности, фраза «сложность алгоритма есть $O(F(n))$ » означает, что при больших значениях n время работы алгоритма (или общее количество операций) не более чем $C * F(n)$, где C – некая положительная константа, n – объем входной информации алгоритма. Если утверждается, что сложность алгоритма $O(n)$, то это означает, что время работы алгоритма растет линейно с ростом n . Запись $O(1)$ означает, что время работы алгоритма не зависит от n .

» Хэш-код

Хэширование – это преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются хэш-функциями, а их результаты называют хэшем или хэш-кодом. Функция хэширования должна обладать следующим свойством: если два объекта считаются равными, то их хэш-коды должны быть одинаковыми; для неравных объектов хэш-

коды могут быть как одинаковыми, так и разными. Ситуация, когда два разных объекта имеют одинаковые хэш-коды, называется коллизией. Хорошая хэш-функция должна давать как можно меньше коллизий. Очевидно, что для достижения этой цели результат хэш-функции должен быть равномерно распределен на всем множестве битовых строк фиксированной длины. Например, если хэш-функция возвращает для всех объектов одно и то же значение, то ее вряд ли можно назвать хорошей.

» Хэш-таблица

Хэш-таблица – это структура данных, реализующая интерфейс ассоциативного массива, а именно, она позволяет хранить пары (ключ, значение) и выполнять три операции: операцию добавления новой пары, операцию поиска и операцию удаления пары по ключу. Хэш-таблица содержит некоторый массив H , служащий для хранения пар ключ – значение. Выполнение операции в хэш-таблице начинается с вычисления хэш-функции от ключа. Вычисленный хэш-код является индексом в массиве H (обычно одному индексу в массиве H соответствует некоторый диапазон хэш-кодов, поэтому более правильно говорить, что по хэш-коду определяется индекс в массиве H). Затем выполняемая операция

(добавление, удаление или поиск) перенаправляется объекту, который хранится в соответствующей ячейке массива H .

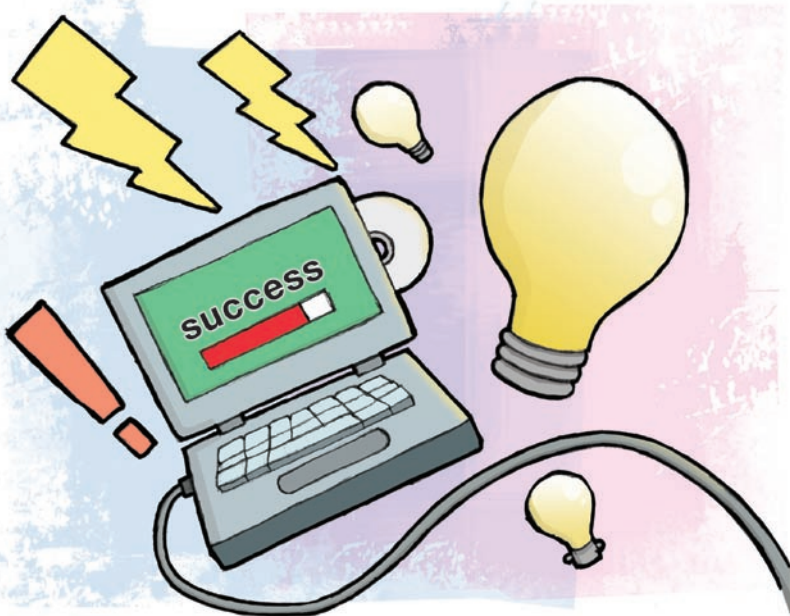
Ситуация, когда для различных ключей получается один и тот же индекс, называется коллизией. Такие события не так уж и редки – например, при вставке в хэш-таблицу размером 365 ячеек всего лишь 23-х элементов вероятность коллизии уже превысит 50 % (если каждый элемент может равновероятно попасть в любую ячейку). Поэтому механизм разрешения коллизий – важная составляющая любой хэш-таблицы.

Число хранимых элементов, деленное на размер массива H , называется коэффициентом заполнения хэш-таблицы [load factor]. Это важный параметр: от него зависит среднее время выполнения операций.

Важное свойство хэш-таблиц состоит в том, что, при некоторых разумных допущениях, все три операции (поиск, вставка, удаление элементов) в среднем выполняются за время $O(1)$. Но при этом не гарантируется, что время выполнения отдельной операции мало. Это связано с тем, что при достижении некоторого значения коэффициента заполнения необходимо осуществлять перестройку индекса хэш-таблицы: увеличить значение размера массива H и заново добавить в пустую хэш-таблицу все пары.

Boost: Набор

Их использование существенно облегчает написание и чтение кода, считает **Семен Есилевский**.



Из чисел в строки и наоборот

» Boost.lexical_cast

Рутинная задача преобразования числа в строку или строки в число решается стандартными средствами C++ на удивление неуклюже. Можно использовать функции `Si atoi()` и `itoa()`, но они работают со строками с стиле Си, а не с объектами типа `string`. Можно использовать класс `stringstream`:

```
#include <sstream>
...
double v = 3.14;
string str;
// Преобразовать число в строку
stringstream ss;
ss << v;
str = ss.str();
// Преобразовать строку в число
str = "100.3";
ss.str(ss);
ss >> v;
```

Если нужно провести много преобразований, то использование объекта `stringstream` оправданно, но в обычном сценарии вводить промежуточную переменную для конвертации одного числа неудобно. Именно для таких случаев и создана шаблонная функция `lexical_cast`:

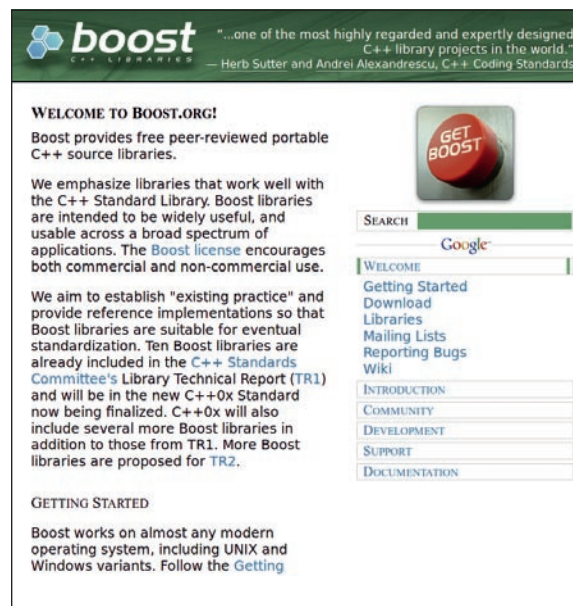
```
#include <boost/lexical_cast.hpp>
...
double v = 3.14;
```

Вступление

Ни для кого не секрет, что язык C++, оставаясь самым мощным компилируемым языком общего назначения, выглядит довольно неудобным по современным меркам. Синтаксис многих конструкций очень запутан, а простые вещи зачастую реализуются неоправданно сложно. Большинство этих проблем ликвидирует набор библиотек **Boost** (<http://www.boost.org/>), призванный, как следует из названия, кардинально повысить продуктивность программирования на C++. **Boost** имеет «полуофициальный» статус, поскольку многие его разработчики являются членами комитета стандартов C++, а некоторые из входящих в **Boost** библиотек уже включены в новый стандарт C++0x. Кроме того, лицензия Boost Software License позволяет свободно и бесплатно использовать **Boost** как в открытых, так и в коммерческих проектах.

Библиотеки **Boost** – чрезвычайно мощные, используют новейшие технологии программирования (такие как шаблонное мета-программирование) и тщательно тестируются, однако назвать их дружественными к программисту трудно. Многие библиотеки имеют излишне запутанный синтаксис при явно недостаточной документации. Особенно это относится к самым мощным, но и самым сложным библиотекам, таким как генератор парсеров *Spirit*. В то же время в повседневной работе чаще всего нужны относительно небольшие, простые и практичные библиотеки из набора **Boost**. Некоторые из этих совсем не страшных и очень полезных библиотек рассмотрены в этой статье. Все они являются «заголовочными» [header-only], поэтому не нужно добавлять к программе что-либо на стадии компоновки.

Эта статья ориентирована на читателей, имеющих некоторый опыт программирования на C++ и хотя бы поверхностно знакомых со стандартной библиотекой и контейнерами STL.



» Домашняя страница проекта Boost. Текст вверху гласит: «одна из самых уважаемых и мастерски разработанных библиотек для C++ в мире», и это чистая правда.



Наш эксперт

Семен Есилевский
Научный сотрудник, вычислительный программист и энтузиаст Linux и Open Source.

библиотек C++

```
string str;
// Преобразовать число в строку
str = boost::lexical_cast<string>(v);
// Преобразовать строку в число
str = "100.3";
v = boost::lexical_cast<double>(str);
    Как говорится, проще не бывает.
```

Заполнение контейнеров

» Boost.assign

Стандартные контейнеры STL имеют один досадный недостаток: чтобы наполнить их элементами, всегда приходится либо писать цикл, либо вручную вызывать **push_back**:

```
vector<int> values;
// Заполняем вектор квадратами индексов
for(int i=1; i<=5; ++i){
    values.push_back(i*i);
}
// Список слов известной фразы Гамлета
list<string> words;
words.push_back("to");
words.push_back("be");
words.push_back("or");
words.push_back("not");
words.push_back("to");
words.push_back("be");
```

Ничего сложного, но избыточность этого кода видна невооруженным глазом. С помощью **Boost.assign** можно избавиться от этой проблемы, используя перегруженный оператор "+=":

```
#include <boost/assign.hpp>
...
values += 1,4,9,16,25;
words += "to","be","or","not","to","be";
```

С ассоциативными контейнерами удобно использовать перегруженный оператор "+" и функцию **insert**:

```
map<string,int> months;
insert( months )
    ("january", 31 ) ("february", 28 )
    ("march", 31 ) ("april", 30 )
    ("may", 31 ) ("june", 30 )
    ("july", 31 ) ("august", 31 )
    ("september", 30 ) ("october", 31 )
    ("november", 30 ) ("december", 31 );
```

Наконец, можно инициализировать любой контейнер сразу при его объявлении с помощью функции **list_of**:

```
vector<int> vals = list_of(1)(4)(9)(16)(25);
```

Для ассоциативных контейнеров предусмотрен особый вариант **map_list_of**:

```
map<int,int> next = map_list_of(1,2)(2,3)(3,4)(4,5)(5,6);
```

В **Boost.assign** есть множество других полезных возможностей – например, заполнение с повторами и заполнение без избыточного копирования данных.

Имитация конструкции foreach

» Boost.foreach

Предположим, у вас есть какой-то контейнер – например, список строк типа **std::list<string>**; и вы просто хотите вывести этот список на экран. «Штатное» решение выглядит так:

```
#include <boost/assign.hpp>
using namespace std;
list<string> lst;
...
// Выводим список
list<string>::iterator it;
for(it=lst.begin(); it!=lst.end(); it++){
    cout << *it << endl;
}
```

Вроде бы все хорошо, но для такой тривиальной операции приходится писать слишком много служебного кода. Итераторы – очень мощное средство, но в данном случае их применение – стрельба из пушки по воробьям. В нашем примере итератор нужен всего лишь чтобы получить значение текущего элемента контейнера. Во многих языках, таких как, например, C# или Java, есть конструкция **foreach**, предназначенная именно для легкого итерирования по любой последовательности или контейнеру. Макрос **BOOST_FOREACH** предназначен для ее эмуляции в C++:

```
// Выводим список
BOOST_FOREACH(string s, lst){
    cout << s << endl;
}
```

Согласитесь, выглядит намного понятнее и проще. Если нужно не просто читать элементы списка, но и модифицировать их, то достаточно объявить переменную цикла как ссылку:

```
// Список чисел
list<double> v = list_of(1)(2)(3)(4)(5);
// Превращаем его в список квадратов чисел
BOOST_FOREACH(double& d, v){
    d = d*d;
}
```

Макрос **BOOST_FOREACH** реализован так, что никогда не выделяет память динамически, а получающийся код по эффективности не уступает написанному вручную с помощью итераторов. Работает он с любыми контейнерами STL, обычными массивами, строками в стиле Си или объектами **string**. Можно также передать **std::pair**, содержащий пару любых итераторов, и макрос «пробегит» диапазон между ними. В теле цикла можно использовать обычные операторы **return**, **continue** и **break**:

```
std::deque<int> deque_int( /*...*/ );
int i = 0;
BOOST_FOREACH( i, deque_int )
{
    if( i == 0 ) return;
    if( i == 1 ) continue;
    if( i == 2 ) break;
}
```

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Циклы могут быть вложенными на любую глубину. Можно итерировать и в обратном порядке, используя аналогичный макрос **BOOST_REVERSE_FOREACH**.

Начав использовать **BOOST_FOREACH**, отказаться от него очень сложно: количество служебного кода зачастую уменьшается в разы, а его читаемость существенно возрастает.

Функции обратного вызова

» Boost.function и Boost.bind

Функции обратного вызова [callbacks] – очень распространенная идиома программирования. Особенно часто они используются при программировании GUI-приложений для отклика на события. Как правило, их реализуют с помощью указателей на функции. Одна беда – синтаксис указателей на функции в C++ просто устаревший! Для иллюстрации создадим заведомо бесполезный класс, выполняющий сложение и вычитание:

```
class Math {
public:
    double do_add(double a, double b){ return a+b; }
    double do_sub(double a, double b){ return a-b; }
};
```

Посмотрим, как можно вызвать эти методы с помощью указателей на функцию:

```
int main(int argc, char* argv[]){
    Math m;
    double (Math::*ptr_add)(double, double) = &Math::do_add;
    double (Math::*ptr_sub)(double, double) = &Math::do_sub;
    cout << "Сумма: " << (m.*ptr_add)(200.0,100.0) << endl;
    cout << "Разность: " << (m.*ptr_sub)(200.0,100.0) << endl;
}
```

И как вам такой кошмарный синтаксис? **ptr_add** не привязан к конкретному экземпляру класса **Math**, и при вызове приходится явно писать **m.*ptr_add**. Выглядит так, как будто у объекта есть метод **ptr_add**; но его нет, и смысл конструкции совершенно иной. Предположим, теперь мы хотим создать функцию, которой будут передаваться два числа и указатель на метод, который к ним надо применить (т.е. собственно классический вариант обратного вызова). Понять, как это записать, очень сложно, и я даже не буду приводить этот код. Одним словом, синтаксис ужасен, а в более сложных случаях он становится вообще практически нечитаемым. На помощь приходит тандем **Boost.function** и **Boost.bind**. С их помощью наш пример становится значительно проще для понимания:

```
#include <boost/bind.hpp>
#include <boost/function.hpp>
double do_operation(boost::function<double(double, double)>
func, double a, double b){
    return func(a,b);
}
int main(int argc, char* argv[]){
    Math m;
    boost::function<double(double, double)> =
        boost::bind(&Math::do_add,&m,_1,_2);
    boost::function<double(double, double)> ptr_sub =
        boost::bind(&Math::do_sub,&m,_1,_2);
    cout << "Сумма: " << ptr_add(200.0,100.0) << endl;
    cout << "Разность: " << ptr_sub(200.0,100.0) << endl;
}
```

Конструкция **boost::function<double(double, double)> func** читается естественным образом – понятно, что **func** – это функция с сигнатурой **double(double, double)**. Этот же тип имеют и переменные **ptr_add** и **ptr_sub** – сразу ясно, что функция **do_operation** готова с ними работать.

Смысл конструкции **boost::bind(&Math::do_add,&m,_1,_2)** понять немного сложнее. **bind** связывает функцию и ее аргументы в единый «вызываемый» объект, обращаться с которым можно как с обычной функцией. В нашем случае мы связываем метод **Math::do_add** с экземпляром нашего класса **m** и «магическими» переменными **_1** и **_2**. При вызове созданного объекта **_1** автоматически заменяется первым фактическим аргументом, **_2** – вторым и т.д. Это так называемые заполнители [placeholders] для аргументов. Наконец, мы просто вызываем **ptr_add** и **ptr_sub** как обычные функции, поскольку они уже связаны с нужным экземпляром класса **Math**.

Научившись работать с **Boost.function** и **Boost.bind**, можно навсегда забыть о кошмарном синтаксисе указателей на функции.

Сигналы и слоты

» Boost.signals2

Если вы работали с библиотекой *Qt*, то знаете, насколько удобной во многих случаях является концепция сигналов и слотов. Сигналы – это, по сути, обобщение идеи функций обратного вызова. Сигнал может быть соединен со множеством слотов, и все они будут вызываться при активации сигнала. Соединение со слотом создается вручную, но разрывается автоматически, когда разрушается вызывающий либо обрабатываемый сигнал объект. Это очень важное свойство – оно позволяет не заботиться о том, что случайно будет вызван слот несуществующего объекта и вся программа «рухнет». Обычные функции обратного вызова никаких гарантий на этот случай не дают.

В **Boost** есть две очень похожие библиотеки, реализующие сигналы – **signals** и **signals2**. Последняя удобнее и современнее, поскольку является полностью заголовочной и позволяет посылать сигналы между разными нитями в многопоточной программе.

Для примера напишем программу, печатающую результат четырех арифметических действий с числами:

```
#include <boost/signals2.hpp>
void mul(float x, float y) { cout << x * y << endl; }
void div(float x, float y) { cout << x / y << endl; }
void add(float x, float y) { cout << x + y << endl; }
void sub(float x, float y) { cout << x - y << endl; }
int main(int argc, char* argv[]){
    boost::signals2::signal<void(float, float)> sig;
    sig.connect(&add);
    sig.connect(&sub);
    sig.connect(&mul);
    sig.connect(&div);
    sig(10,5);
}
```

Мы создали сигнал с нужной нашим функциям сигнатурой **void(float, float)**, соединили его со всеми четырьмя функциями методом **connect()** и активировали, передав параметры 10 и 5. В результате вызываются все функции поочередно.

Сигналы можно соединять не только с функциями, но и с любыми вызываемыми объектами (для которых определен оператор **()**), в том числе с теми, которые создает **boost::bind**. Например, в нашем примере с функциями обратного вызова можно было бы написать

```
boost::signals2::signal<double(double, double)> sig;
sig.connect( boost::bind(&Math::do_sub,&m,_1,_2) );
cout << *sig(200,100) << endl;
```

При этом вызывается нужный метод, но значение возвращается в виде указателя, и его нужно разыменовать.

Если вызывается несколько сигналов, которые возвращают значения, то результат такого вызова неоднозначен. Что при этом

будет возвращено, решает пользователь с помощью так-называемых «комбинаторов» [combiners] – специальных объектов, которые аккумулируют значения, возвращенные всеми слотами, и обрабатывают их. Однако это уже «высший пилотаж». Если же игнорировать возвращаемые значения слотов, то пользоваться сигналами чрезвычайно просто.

Гетерогенные контейнеры

» Boost.variant и Boost.any

Один из хрестоматийных вопросов, постоянно задаваемых на тематических форумах – как создать гетерогенный контейнер в C++? C++ – строго типизированный язык, и в массивах или контейнерах STL можно хранить значения только какого-то одного определенного типа. В то же время часто возникает необходимость создать гетерогенный контейнер, содержащий, скажем, одновременно числа и строки. Сделать это силами стандартной библиотеки можно, но решение будет громоздким и небезопасным, т.к. придется использовать «тяжелую артиллерию» вроде указателей типа `*void`.

В **Boost** есть средства, позволяющие создать удобные и безопасные гетерогенные контейнеры со строгой проверкой типов. Начнем со случая, когда мы четко знаем, что будем хранить либо строки, либо целые числа, либо числа с плавающей точкой:

```
#include <boost/variant.hpp>
using namespace std;
...
typedef boost::variant<string, double, int> data_t;
vector<data_t> data;
data.push_back(123);
data.push_back(3.14);
data.push_back("Hello!");
```

Мы перечисляем все нужные типы как параметры шаблонного типа **variant**, после чего можем использовать его как тип для нашего контейнера. В контейнер теперь можно добавлять данные всех перечисленных типов. Чтобы прочитать данные, нужно либо точно знать тип текущего элемента (что бывает редко), либо действовать методом проб и ошибок. Например, так можно вывести все данные из нашего контейнера вместе с их типом:

```
BOOST_FOREACH(data_t& item, data){
    int* ptr1 = boost::get<int>(&item);
    if(ptr1) cout << "Это целое число." << *ptr1 << endl;
    double* ptr2 = boost::get<double>(&item);
    if(ptr2) cout << "Это число с плавающей точкой." << *ptr2 << endl;
    string* ptr3 = boost::get<string>(&item);
    if(ptr3) cout << "Это строка." << *ptr3 << endl;
}
```

Шаблонная функция `boost::get<>` пытается получить значение заданного типа из переменной типа **variant** (передается ее адрес).

Если это удастся, она возвращает указатель нужного типа на это значение, а если не удастся, то **NULL**. Перебирая все варианты, можно определить и тип элемента, и его значение. Есть и другой вариант этой функции, который принимает не адрес, а саму переменную, и возвращает не указатель, а само значение. Если тип не совпадает, то генерируется исключение типа `bad_get`:

```
try {
    int val = boost::get<int>(item);
} catch(const boost::bad_get&){
    cout << "Это не целое число!" << endl;
}
```

Другой сценарий использования гетерогенного контейнера возникает, когда нужно хранить значения действительно любого типа либо когда вариантов типов очень много. В таком случае на помощь приходит **Boost.any**:

```
boost::any data; // Можно хранить что угодно!
...
// Пытаемся извлечь строку
try{
    cout << any_cast<string>(data) << endl;
} catch(const boost::bad_any_cast&) {
    cout << "Это не строка!" << endl;
}
```

Функция `any_cast` ведет себя точно так же, как и `boost::get`, и тоже существует в двух вариантах, возвращающих указатель либо само значение.

Выводы

Библиотеки **Boost** вполне оправдывают свое название – они позволяют повысить продуктивность программирования на C++ и создают удобства, невиданные в рамках базового языка и стандартной библиотеки. Особенно хорошо с этой задачей справляются «маленькие» библиотеки вроде **Boost.foreach** и **Boost.assign** – простые, легкие в освоении и имеющие удобный синтаксис. Они заполняют пробелы стандартной библиотеки и исправляют недостатки синтаксиса самого языка. Ярким примером являются **Boost.bind** и **Boost.function**, позволяющие забыть об указателях на функции, их ограничениях и ужасном синтаксисе. Однако не все рассмотренные в этой статье библиотеки являются «маленькими». Например, **Boost.signals2** имеет свои «темные углы», а синтаксис комбинаторов способен повергнуть новичка в уныние. В этом особенность **Boost** – отход от простых моделей использования часто приводит пользователя в плохо документированные «дебри». Тем не менее, **Boost** – обязательная часть арсенала современного программиста на C++, который хочет работать действительно эффективно, а не бороться постоянно с синтаксическими тонкостями и излишней низкоуровневостью этого языка. **LXF**

Что осталось за кадром

В **Boost** входит множество библиотек, и многие из них вполне могут пополнить список простых в использовании и очень полезных инструментов для эффективного программирования. Приведу краткие описания некоторых из них.

» **Boost.array** – удобная обертка вокруг стандартных массивов фиксированной длины, позволяющая использовать их так же, как обычные контейнеры STL, и инициализировать «на лету»:

```
boost::array<int,3> a = { 1, 2, 3 };
```

» **Boost.multi_array** – библиотека для удобной работы с многомерными массивами фиксированного размера. Позволяет забыть о мучениях с выделением памяти и «многозатяжных» указателях, без которых не обойтись

при использовании и обработке стандартных многомерных массивов:

```
// Трехмерный массив с размерами 3x4x2
boost::multi_array<double, 3> A(boost::extents{3}[4][2]);
A[0][1][2] = 3.14; // Обычный доступ к элементу
```

Синтаксис местами довольно неуклюжий, но удобство использования перекрывает эти недостатки.

» **Boost.optional** – тип данных, способный либо хранить значение заданного типа, либо быть пустым. Может использоваться во многих ситуациях вместо дополнительного флага, сигнализирующего о том, что объект пуст.

» **Boost.ptr_container** – контейнер указателей на динамически созданные объекты, который является их «хозяином» и автоматически удаляет их, когда разру-

шается сам. Очень удобен для хранения, например, полиморфных объектов, связанных иерархией наследования.

» **Boost.random** – библиотека для генерации случайных чисел. Позволяет получать разнообразные статистические распределения и использовать различные алгоритмы. Для простых случаев синтаксис излишне сложен, но зато не нужно каждый раз самостоятельно преобразовывать данные стандартного случайного генератора.

» **Boost.string_algo** – набор алгоритмов для работы со строками как с обычными STL-контейнерами. Удачно дополняет класс `string`, но, к сожалению, имеет невразумительную документацию.

Gnome 3: Как его

Shell вызвал затруднения? **Джонатан Робертс** собрал средства и методы для восстановления утраченных функций, чтобы получить рабочий стол мечты.



Наш эксперт

Джонатан Робертс – рукодельник, который пользовался Gnome Shell задолго до его официального выхода.

Похоже, история проекта Gnome развивается по спирали. Девять лет назад релиз версии 2.0 стал заметным событием из-за его стремления к рационализации рабочего стола: разработчики убрали «невнятные» опции и создали Указания к интерфейсу, стараясь обрести гармоничный рабочий стол, а не просто расхлябанное нагромождение модулей.

Прокрутим почти десятилетие, и что мы имеем? Релиз Gnome 3.0 снова примечателен своей попыткой рационализировать рабочий стол. Многие опции убраны, а на смену им пришли новые интерфейсы и шаблоны взаимодействия.

Как и в случае с Gnome 2, многие пользователи отреагировали с гневом и разочарованием. И вообще, если надо раскрасить рабочий стол в желто-зеленый, под цвет формы вашей любимой футбольной команды, то как это теперь делается?

В те времена серия Gnome 2 многие опции постепенно вернула, и получилось одно из самых гибких и популярных окружений рабочих столов. А что произойдет с Gnome 3? Вызовет ли он кровоизлияние у пользователей, или проследует эволюционным путем, как его предшественник?

На самом деле, определенная эволюция нужна – например, требуется более хорошая документация; опции настройки имеются в изобилии, просто надо знать, где искать. Данный урок покажет вам средства и методы восстановить ваш знакомый ход работы, настроить вид рабочего стола и расширить его так, как и не снилось разработчикам.

Мы сделаем упор на новый интерфейс Shell – интерфейс по умолчанию, если ваш компьютер его поддерживает, и нашей первой задачей станет восстановление некоторых функций, которых не хватает в Gnome 3 по сравнению с Gnome 2 и другими современными рабочими столами. Чаще всего, возможно, упоминается панель задач, которая использовалась для отображения в низу экрана запущенных программ. В Gnome Shell она заменена на режим Overview, который надо смотреть отдельно.

Изменение, может, и небольшое, но многие нашли его выбивающим из колеи. Ведь панель задач была общей чертой почти всех компьютерных платформ на протяжении более десятилетия, а от старых привычек избавляться непросто.

Однако есть несколько приложений третьих сторон, которые не только повторяют функции панели задач, но также и отлично настраиваются и сочетаются с темой Gnome 3, стоящей по умолчанию.

Клоны панели задач

Возможно, лучшая имитация традиционной панели задач – *Tint 2*. Он очень легковесный, отнимает всего 5 МБ резидентной памяти при обычных операциях, и имеет широкий спектр опций настройки: можно задать, чтобы панель прилегалась к любому краю экрана, сделать ее прозрачной и даже содержать системный лоток (для значков программ вроде *Banshee*).

Все эти установки можно поменять с помощью программы *tint2conf*, или отредактировав файл *tint2rc* в `~/.config/tint2`. Если повозившись с *Tint 2*, вы поймете, что он в вашем стиле, можно обеспечить его автоматический запуск при каждом входе, с помощью программы *gnome-session-properties*. Хотя ее не видно в режиме Overview и она не входит в новый Gnome Control Center, ее можно запустить из терминала или из диалога запуска (Alt+F2).

Если вы предпочли бы док рабочего стола в стиле Mac, то будете рады узнать, что Docky работает в Gnome Shell не хуже, чем в Gnome 2. Простой установки из репозитория дистрибутива

достаточно, чтобы она появилась в списке программ по умолчанию.

Вероятно, вам уже полегчало, однако верхняя панель, зияющая отсутствием меню с программами или списка лег-

ко доступных избранных, все еще вызывает раздражение.

К счастью, Рон Йорстон [Ron Yorston] создал расширение (см. врезку о расширениях), которое восстанавливает эти функции в верхней панели, а также дает возможность перенести часы

«Похоже, история проекта Gnome развивается по спирали.»



» *Tint 2* и *Docky* восполняют в Gnome 3 нехватку утраченной нижней панели.

ВСКРЫТЬ

на прежнее место, направо, и выключить динамическое создание рабочих полей.

Его пакет можно скачать с <http://intgat.tigress.co.uk/rmy/extensions/index.html>. Это удобный архив TGZ, и для установки нужно просто распаковать его в домашнюю папку: `tar -xvzf gnome-shell-frippery-0.0.2.tgz`. Для активации нового расширения перезапустите Gnome Shell, вызвав нажатием Alt+F2 диалог запуска, с последующим вводом `restart`. Тогда изменения станут видны, но при этом кое-что следует знать.

Избранное просто отражает содержимое вашего Dash (док-подобной панели в режиме Overview) – чтобы его поменять, сделайте нужные изменения в Dash.

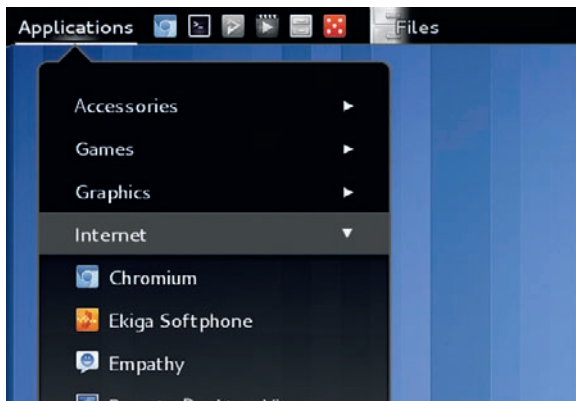
Учтите также, что при отключении динамических рабочих полей вам придется задать количество используемых рабочих полей. Это делается в программе `gconf-editor` – нужная установка лежит в Apps > Metacity > General > Num_Workspaces. Без этого вам останется только одно рабочее поле, и вы лишитесь одной из лучших возможностей Gnome.

Восстановив большую часть важных и недостающих многим функций, можно заняться настройкой прочих имеющихся возможностей.

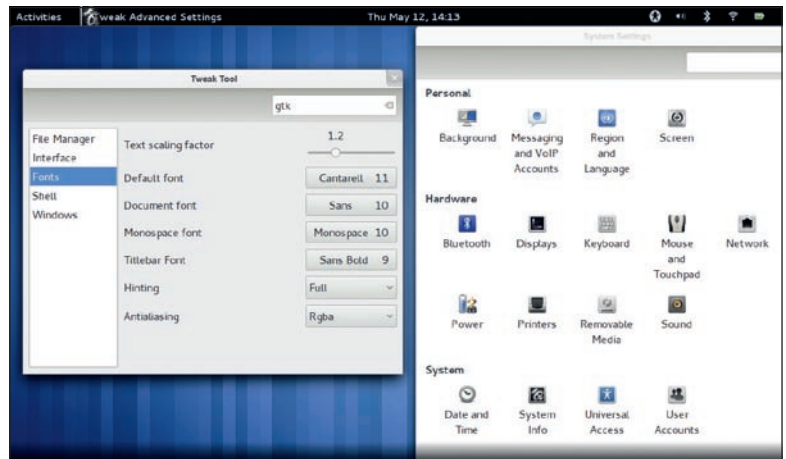
Хотя эти настройки зачастую невысокого полета – вроде изменения размера шрифта или действия при закрытии крышки ноутбука, многие пользователи считают их данностью, и это может украсить или загубить опыт работы с компьютером.

В новом Control Centre в Gnome 3 не включены многие из этих опций, и вам придется задействовать `gnome-tweak-tool`. Во многих дистрибутивах с Gnome 3 `gnome-tweak-tool` тоже доступен, так что установите его через свой менеджер пакетов.

По сути, это хранилище большинства важнейших опций, не попавших в новый Control Center Gnome 3. Там вы найдете все вышеупомянутые, наряду с многими другими. Есть панель поиска – она



» Панели по умолчанию не хватает кучи функций, но мы нашли расширение для восстановления ее полезности.



написана в `GTK 3` и гармонично сочетается с остальным рабочим столом. Уделите время на изучение `gnome-tweak-tool`; но если и там вы не обнаружите некой настройки, которая, по вашему мнению, где-нибудь да должна присутствовать, у вас останется два места для поиска.

С тех пор как Gnome сосредоточился на рационализации интерфейсов пользователя, опции, которые, по мнению разработчиков, усложняли диалог настройки, не всегда полностью удалялись, а переносились в скрытую базу данных.

Таких баз данных две, с тех пор как Gnome 3 отметил начало перехода от старого `gconf` к новой системе `gsettings`. В конечном итоге, все установки Gnome и приложений должны осесть в `gsettings`, но сейчас на наличие доступных опций нужно проверять обе.

Обе базы снабжены простыми графическими приложениями, позволяющими исследовать и менять все доступные опции. Это `gconf-editor` и `dconf-editor`, соответственно. Обратите внимание, что `gconf-editor` запускается обычным способом, а вот для запуска `dconf-editor` нужен диалог `run` или терминал.

Также следует упомянуть, что Control Centre и `gnome-tweak-tool` всего-навсего отображают базы данных в простом, урезанном диалоге для пользователей. Если вы собираетесь проделать какие-либо изменения в базе данных, позаботьтесь запомнить исходное значение, чтобы можно было восстановить его, если вы передумаете.

Понимание темы

Может вы удивитесь, а может и нет, но настройка тем в Control Centre тоже отсутствует, и снова пробел заполняет `gnome-tweak-tool` – или `gconf` с `gsettings`, как вам больше нравится.

Вы, скорее всего, уже обратили внимание на опцию `theme` в `gnome-tweak-tool` и были удивлены или сбиты с толку их количеством. Откуда их столько и зачем они?

» `gnome-tweak-tool` и новый Control Centre в Gnome 3: вместе они составляют мощную комбинацию.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Темы значков и курсоров говорят сами за себя, их мы опустим. Тема *GTK* задает стиль всем виджетам – то есть кнопкам, вкладкам, меню и полосам прокрутки, формирующим управление всех ваших программ.

В Gnome 3, темы *GTK* идут в двух вариантах: **gtk-3.0** и **gtk-2.0**. Они связаны с двумя различными версиями *GTK*, и каждый применим только в программах, написанных в данной версии. То есть пока все ваши программы не портированы в *GTK 3*, придется вам искать в обеих версиях сходные темы, если вы хотите, чтобы рабочий стол выглядел опрятно.

Тема *Shell* – новинка Gnome 3: она определяет вид интерфейса *Shell* – то есть вид верхней панели и всего, что содержится в режиме *Overview*, доступном по нажатию на клавишу *Windows*. И наконец, *Current Theme* в разделе *Windows* относится к тому, что раньше называлось темой *Metacity*. Там определяется вид окон, содержащих ваши приложения: их заголовки, границы и управляющие элементы для закрывания, сворачивания и распахивания.

Набором тем сначала следует обзавестись. Многие доступны в репозиториях дистрибутива – проверьте сначала там. Если вам встретится тема, недоступная в репозитории, установите ее вручную, распаковав ее содержимое в определенный каталог.

» **GTK** и *Window* темы можно установить либо в `~/themes/$theme_name/` (для одного пользователя), либо в `/usr/share/themes/$theme_name/` (для всех пользователей). Если вы устанавливаете тему *GTK*, ее нужно разместить либо в каталоге **gtk-3.0**, либо в **gtk-2.0**, а если тема *Metacity*, то в каталоге **metacity-1**.

» Темы значков и курсоров можно установить в `~/icons` или `/usr/share/icons.w`.

При установке новой темы для Gnome Shell сперва убедитесь, что установлено расширение *user-theme*. В большинстве дистрибутивов оно входит в пакет **gnome-shell-extensions**, а его исходник можно скачать с <http://live.gnome.org/GnomeShell/Extensions>: там также имеются инструкции по сборке и установке.

Установив расширение, вы сможете скопировать темы Gnome Shell в `~/themes/$theme_name`, поместив их в каталог **gnome-shell**, а затем выбирать их в *gnome-tweak-tool*.

Мы уже освоили много нового, исправив ряд самых неприятных неудобств в Gnome Shell и частично вернув гибкость Gnome 2.

Уже кое-что, но еще не все. Иногда вам требуется взять дело в свои руки, создав собственную идеальную тему или расширение с небольшой, но для вас решающей функцией. К счастью, главной целью разработчиков Gnome 3 было как раз упрощение таких настроек, чтобы они стали доступнее для широких масс.

Далее следует ускоренный курс по новым возможностям создания тем в Gnome 3, где излагаются основы *CSS* и как применять их к *GTK 3* и *Gnome Shell*.

Мы были бы рады исследовать возможности расширения *Gnome Shell*, но пока дела обстоят так, что в системе почти нет документации, а *API* имеет склонность меняться.

Если вам интересно изучить это самостоятельно, рекомендуем в качестве прекрасного бесплатного введения *Eloquent Javascript*. Когда вы освоите основы языка, обзор информации и кода в <http://live.gnome.org/GnomeShell/Extensions> поставит вас на ноги.

В вашем стиле

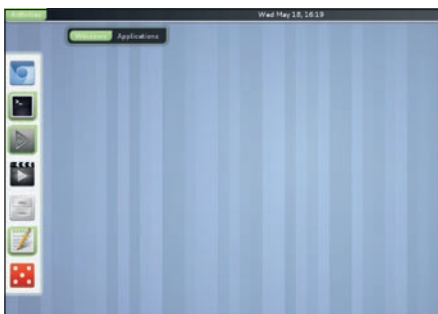
CSS означает «каскадные таблицы стилей [Cascading Style Sheets]»; это механизм добавления информации о стиле, такой как шрифты, цвета и позиция элемента в структурированном документе. Шире всего они применяются в Интернете, но разработчики Gnome 3 воспользовались данным синтаксисом для тем *GTK 3* и *Gnome Shell*.

Синтаксис таблиц стилей довольно простой. Каждая таблица состоит из набора «правил», а каждое из правил содержит «селектор» и «блок объявления».

В блоке объявления перечислено некоторое число свойств и им присвоены значения, например, `'color: red;'`. Обратите внимание, что свойство и значение разделены двоеточием, а присваивание заканчивается точкой с запятой. Также блоки объявления заключаются в фигурные скобки.

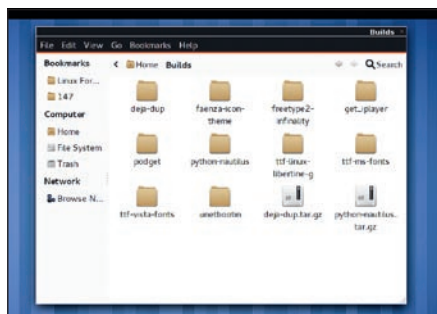
Селектор определяет, к какому именно элементу или виджету применяется содержимое блока объявления: например, к виджету **GtkComboBox** или к элементу параграфа HTML. Виджет *GTK* можно задавать как обобщенный тип (**GtkComboBox**), по имени (**#button1**), или как класс, определяющий группу виджетов (**.button**), применяемых к флажкам, переключателям, стандартным кнопкам и кнопкам-переключателям.

Три лучшие темы



1 Gaia

Легкая и живая тема, приятный контраст с псевдо-промышленной темой по умолчанию. Ее создал Half-left с DeviantART, плодотворнейший автор тем для Gnome Shell на сегодняшний день. www.half-left.deviantart.com/



2 Elegant Brit

Еще одна, тоже приятная тема. Преимущество ее в том, что она содержит темы *GTK 3* и *Window*, приводя к гармоничному виду рабочего стола. Доступна на отличном сайте, где можно отыскать и другие темы. www.gnome-look.org



3 Faenza

Полный набор квадратных значков для рабочего стола Gnome, вдохновленных iOS. Набор поставляется с выбором светлых, темных и очень темных значков, чтобы они хорошо смотрелись на любой панели или меню, независимо от их цвета. Доступны либо на *gnome-look*, либо на DeviantART.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Расширения Gnome Shell

Система расширений Gnome Shell предоставляет пользователю способ расширить и изменить рабочий стол по умолчанию, не трогая код, включенный в основную кодовую базу. Расширения – это просто комплекты

Javascript и CSS, способные взаимодействовать с ядром Gnome Shell, чтобы изменять способы обработки окон, добавления или удаления элементов интерфейса и многого другого. Сейчас поддержка управления рас-

ширениями ограничена. Для установки нового расширения скопируйте его вручную в каталог `~/local/share/gnome-shell/extensions`. Для отключения расширения удалите его оттуда.

Простой набор правил будет выглядеть примерно так:

```
GtkComboBox {
    padding: 4;
    color: #ffffff;
}
```

В CSS можно использовать куски синтаксиса других типов, и там есть интересные способы комбинировать их друг с другом и получать сложные выражения, однако это уже хорошо задокументировано в портале разработчиков Gnome. Прежде чем читать дальше, стоит заглянуть на <http://developer.gnome.org/gtk3/stable/GtkCssProvider.html>, в особенности в раздел селекторов и `@rules`.

Итак, вы освоили основы style sheets, и пора написать вашу первую тему. Вместо того, чтобы начать с нуля, лучше взять готовую тему, изменить ее и наблюдать за эффектом во время работы.

Для начала мы взяли Adwaita – это самый полный пример темы GTK 3. Скопируем ее в подкаталог `.theme` вашего домашнего каталога, снабдим новым именем, а потом зайдем внутрь и посмотрим содержимое.

```
cp -r /usr/share/themes/Adwaita ~/.themes/New
ls ~/.themes/New/
```

Вы увидите папку `gtk-3.0`, о которой мы уже говорили. В ней должно быть два CSS-файла: файл `gtk.css` содержит список цветовых определений, а файл `gtk-widgets.css` – список правил для всех виджетов.

Каждый список правил в `gtk-widgets.css` определяет стиль отдельного виджета. Если вы не знаете, какому виджету на экране соответствует имя виджета, вам будет сложно начать редактирование.

Но это легко преодолить: ведь документация разработчиков Gnome содержит список всех виджетов – включая описания и экранные снимки того, что нужно ожидать. Простейший способ добраться до этой информации – от души попользоваться функцией поиска по сайту от Google. Например:

```
site:developer.gnome.org/gtk $widget_name
```

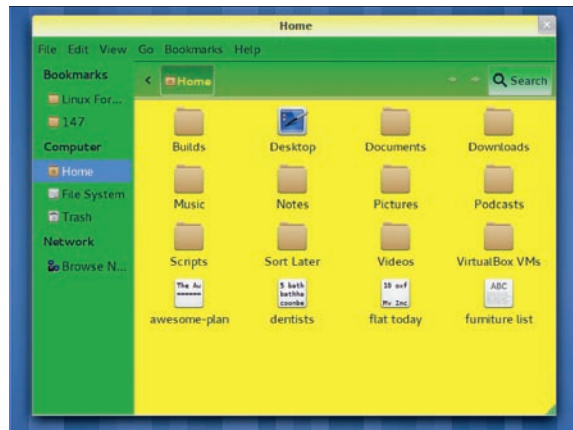
Многие цвета в `gtk-widgets.css` задаются с помощью `@rules`, которые являются просто указателями на определения из файла `gtk.css`. Так что если вы хотите поменять цвет виджета, возьмите его имя из `gtk-widgets.css` и воспользуйтесь функцией поиска текстового редактора, чтобы найти соответствующий пункт в `gtk.css`.

Создаем стиль

Ну вот, мы готовы к редактированию файлов для переделки темы Adwaita на свой лад. Лучше всего следовать пошаговому принципу, проделывая отдельные изменения, сохраняя файл и наблюдая за результатом. Тогда вы сможете быстро отлавливать любые ошибки и осознаете влияние изменения на рабочий стол.

Чтобы изменения стали видимыми, сначала задайте тему, используя `gnome-tweak-tool`. Затем вам нужно тестовое приложение GTK 3, которое надо перезагружать после каждого сохранения, чтобы видеть результат. Для этого сгодится `Nautilus`: в нем полно всяких виджетов, и он быстро открывается и закрывается.

Алгоритм создания новой темы для Gnome Shell весьма похожий. Однако есть и важные отличия:



» Кто хочет отметить переход команды Норич-Сити в премьер-лигу специальной GTK-темой, свяжитесь с автором.

1 Чтобы можно было выбирать разные темы в `gnome-tweak-tool`, вам придется установить расширение `user-theme`, как описано выше.

2 Как и раньше, гораздо проще будет начать с известной темы и менять ее на свой вкус, но знайте, что тема Gnome Shell, стоящая по умолчанию, хранится в другом месте:

```
cp -r /usr/share/gnome-shell/theme/* ~/.themes/New_Shell/gnome-shell/
```

3 Заглянув в каталог, вы увидите, что хотя там много файлов, только один из них – файл CSS: это `gnome-shell.css`. Его синтаксис почти совпадает с тем, что в файле `gtk-widgets.css`, но на сей раз проект Gnome предоставил очень мало документации насчет того, какой селектор соответствует какому элементу экрана. Однако вам не придется применять метод проб и ошибок: предприимчивые ребята с форумов Ubuntu прошлись по файлу, стоящему по умолчанию, и великодушно повставляли комментарии, объясняющие соответствие многих элементов (<http://ubuntuforums.org/showpost.php?p=8997676&postcount=69>). При работе рекомендуем держать это руководство открытым.

4 Чтобы ваши изменения возымели силу, нужно выбрать тему из `gnome-tweak-tool` и перезапустить Gnome Shell после каждого изменения. Это можно сделать, нажав `Alt+F2` запустить диалог `run`, и после этого ввести `r` или `restart`. И снова, правильным будет подход пошаговых изменений.

Если больше ничего не работает...

Вот вам и все. Если, открыв для себя гибкость Gnome 3 и потратив часы на попытки привести его симпатичному для вас виду, вы не согласились на переход, не отчаивайтесь. По крайней мере, вы неплохо провели время, и есть ведь `Xfce`, `KDE`, `LXDE`, `Unity` или любой другой рабочий стол Linux вам на радость.

Мы надеемся, что наш урок дал вам почувствовать вкус возможностей, и еще больше пользователей попробуют Gnome 3. Обидно, что во многих областях нет приличной документации, но мы уверены, что когда уляжется пыль от третьего релиза и внимание переключится на более эволюционные изменения, ситуация улучшится. **Linux**

» **Через месяц** Соберем себе личное облако с помощью *Eucalyptus*.

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru

» В этом месяце мы ответим на вопросы про...

- 1 Gnome 3 в Ubuntu
- 2 Изменение размеров разделов
- 3 OpenSUSE
- 4 Спецсимволы
- 5 Изменение языков
- 6 Проблемы конфигурации
- 7 Управление логическими томами
- 8 Двойную загрузку

1 Ubuntu с Gnome3

В Недавно я обновился до Ubuntu 11.04, пару дней повозился с Unity, и мне не понравилась. Я уже испытал Gnome 3 и считаю эту среду намного удобнее. Но установить Gnome 3 в Ubuntu 11.04 сложно, если вообще возможно. Я попробовал методы, предложенные на askubuntu.com и unixmen.com. Первый из них работал с бета-релизом 11.04, но теперь установка просто прекращается. Второй метод не работает вообще. Теперь я вижу только приглашение к регистрации; система отображает стандартный фон и ничего больше.

Нет ли способа установить Gnome 3 в Ubuntu 11.04, и существуют ли какие-нибудь дистрибутивы кроме Fedora, где Gnome 3 устанавливается по умолчанию?

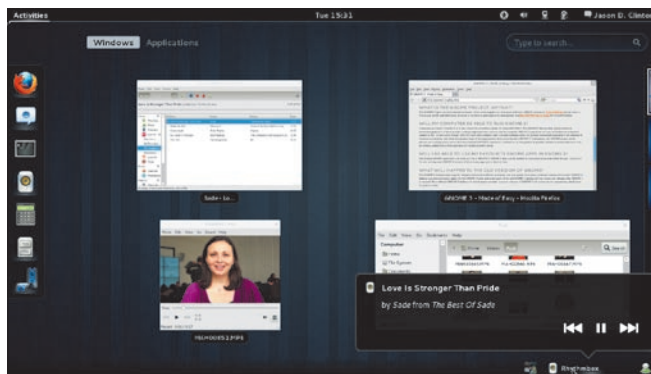
Майк Кэтфорд (Mike Gattford)

О Пока по умолчанию ни один дистрибутив

не включал Gnome 3, кроме Fedora 15, вышедшего в конце мая, потому что все остальные опережали выпуск Gnome 3. В новых циклах можно ожидать Gnome 3. Но многие ли примут Gnome 3 по умолчанию, зависит от реакции на Fedora 15. Впрочем, включение по умолчанию не обязательно – была бы доступность. OpenSUSE уже обзавелся пакетами из стабильного репозитория Gnome на <http://download.opensuse.org/repositories/GNOME:/STABLE:/3.0>, готовыми к установке через *YaST*. Live CD OpenSUSE с Gnome 3 можно взять на <http://www.gnome3.org/tryit.html>.

Вы должны быть в состоянии установить Gnome 3 и в Ubuntu 11.04, добавив должный репозиторий PPA. Это делается через *Synaptic*, но гораздо проще установить обновление командами, данными из терминала:

```
sudo add-apt-repository ppa:gnome3-team/
```



» Если Unity вам не по душе, в Ubuntu 11.04 можно установить Gnome 3.

```
gnome3
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install gnome-shell
```

Третья команда может вернуть сообщение об ошибке, предлагающее запустить ее еще раз с опцией `-f`. На это скамануйте

```
sudo apt-get dist-upgrade -f
```

и продолжайте установку. По завершении выйдите из системы и повторно зарегистрируйтесь, выбрав опцию Gnome Shell, когда экран регистрации GDM появится снова.

Эта процедура требует серьезного предупреждения: откат после установки Gnome 3 – задача долгая и трудоемкая, куда сложнее переустановки. Поэтому, прежде чем браться за эту операцию, позаботьтесь о работоспособной резервной копии, или создайте систему с двойной загрузкой: на втором экземпляре *Natty* можно попробовать выполнить процедуру, ничем не рискуя. **ГМ**

2 Сжатие Windows

В Пытаясь перемасштабировать раздел *Windows*, чтобы установить *OpenSUSE* и создать систему с двойной загрузкой, я запустил программу дефрагментации *Windows* и затем начал установку. При создании разделов мне было предложено создать раздел *Windows* размером 172 Гб, хотя занято было только 20 Гб. Я загрузил стороннюю программу дефрагментации, *PerfectDisk* от *Raxco*, которую мне порекомендовали, запустил ее и дефрагментировал диск – сразу же и потом, после перезагрузки. Но установщик все равно настаивал на уменьшении раздела *Windows* только до 164 Гб. Взглянув на карту кластеров, я решил, что причина в неких метаданных, сидящих посреди диска: *PerfectDisk* не перемещает их, чтобы освободить пространство. Вот эта карта:

Наши эксперты

» Мы найдем ответы на любой вопрос – от проблем с установкой системы или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик

Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Пол Хадсон

Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Валентин Синецын

В редкие свободные минуты Валентин обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.



Майк Сондерс

Майк был одним из создателей прототипа *LXF* – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и *SNES*.



Грэм Моррисон

Когда Грэм не обозревает кучи программного обеспечения и не халтурит с *MythTV*, он готов к ответам насчет любого оборудования и проблем виртуализации.



Юлия Дронова

Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования www.unixforum.org.

Куда посылать вопросы

Пишите нам по адресу answers@linuxformat.ru или спрашивайте на форуме: www.linuxformat.ru

Block 1 - Directory
Blocks 2-18 - free space
Blocks 19-161 - files
Blocks 162-560 - free space
Block 561 - Metadata
Blocks 562-1011 - MFT zone
Blocks 1011-1800 - free space
Blocks 1801-1802 - Metadata

По-моему, хранить в таком большом разделе всего 20 ГБ данных — это расточительство. Не подскажете ли вы, как его уменьшить?

mikejd

Вы не указали исходный размер раздела Windows, но существует предел, до которого можно сжать раздел NTFS за один шаг, вследствие размещения данных в середине файловой системы. Из-за этого установщик может уменьшить размер файловой системы только примерно наполовину. Но приятно то, что в процессе перемасштабирования метаданные перемещаются в новую середину файловой системы. Если Вы согласитесь с предложенным размером и прервете установку раньше, чем создадутся разделы Linux, то сможете запустить установщик повторно, и Вам еще раз предложат уменьшить размер файловой системы Windows примерно вдвое. Перед вторым и последующими масштабированиями неплохо будет загрузить Windows и еще раз дефрагментировать диск.

Хорошей идеей считается блокировка виртуальной памяти в Windows перед запуском дефрагментации, чтобы файл подкачки не мешал этому процессу. Но по завершении не забудьте снова включить виртуальную память. **ПХ**

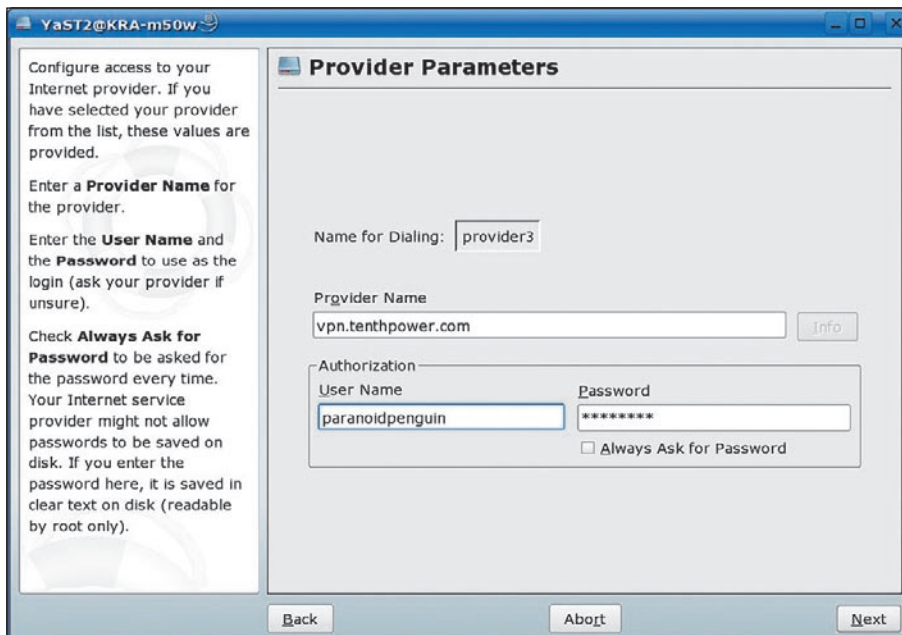
3 Бессвязный OpenSUSE

Я установил OpenSUSE 11.3 с LXF DVD136. Но в этот дистрибутив не входят драйверы HP, необходимые для моего принтера и, по всей видимости, для контроллера модема (Kinternet?). В результате я не могу ни печатать, ни выйти в Интернет.

Необходимые RPM есть на DVD OpenSUSE 11.0 (коробочная версия). Я скопировал RPM-пакеты hplip, hplip-hpijs и Kinternet в новый каталог /rpms и настроил его как репозиторий. Пытаясь установить их с помощью Yast, я получаю сообщение об отсутствующей зависимости, libcrypto.so.0.9.8. При выборе опции Ignore ПО не работает. Коробочная версия OpenSUSE 11.0, похоже, такого файла не содержит. Я предполагаю, что SUSE использует какой-то обходной путь. Но какой?

Хью С. Джонс (Hugh C. Jones)

Дистрибутив OpenSUSE на LXF DVD136 — среда Live CD. Хотя его можно установить на жесткий диск, он ограничен размерами CD и поэтому содержит не все пакеты, имеющиеся на DVD — например, отсутствуют упомянутые Вами. Доступ в Интернет есть (иначе зачем нужен web-браузер?) но графический интерфейс для коммутируемых модемов из состава исключен. Создается впечатление, что частично Вы уже сами решили проблему, получив RPM из другого



Кинтернет не входит в Live CD OpenSUSE или LXF DVD136, но можно подключиться к Интернет из wvdial и установить его.

источника; осталось только установить ПО для модема, а все остальное, кроме библиотеки crypto, можно будет загрузить потом. Библиотека содержится в RPM openssl — вы найдете пакет с нужным файлом на сайтах вроде <http://rpm.pbone.net> или <http://www.rpmfind.net>.

Однако в OpenSUSE, как и в любом другом дистрибутиве, смешивать пакеты из разных версий значит искать неприятностей на свою, э-э, голову. Вы можете добиться подключения Интернета и немедля обновиться до правильной версии, но получить изначально правильную версию будет безопаснее. Отчасти это вопрос «курицы и яйца»: коли нет Интернет-соединения, то и версию получить неоткуда. Но упомянутые сайты содержат ссылки для загрузки (убедитесь, что Вы получаете файлы OpenSUSE 11.3 для i586), и их можно посетить с другого компьютера (допустим, дома или на работе) или из-под другой операционной системы на Вашем компьютере. Спишите файлы на USB-брелок, и Ваша проблема будет решена.

Вы можете установить соединение с Интернет, загрузившись с Live CD, но там нет GUI для этой цели: установлен только набор номера wvdial. Вы можете настроить его следующей командой:

```
wvdialconf ~/.wvdialrc
```

Она обнаружит Ваш модем и создаст файл настройки в Вашем домашнем каталоге. Откройте его в своем любимом текстовом редакторе и вставьте телефонный номер, имя пользователя и пароль — по формату файла понятно, какие изменения и куда требуется внести. Сохраните файл и установите соединение командой

```
wvdial
```

Теперь можете установить GUI номеронабирателя для KDE, под названием KPPP, или Kinternet от OpenSUSE, но при необходимости у Вас всегда будет обходной путь — wvdial. Драйверы для большинства принтеров и сканеров HP находятся в пакете hplip; Вы можете установить их через

Yast сразу же после подключения, не связываясь с потенциально конфликтными версиями из более ранних релизов. **МС**

4 Сложная символика

У меня есть несколько файлов, имена которых содержат спецсимволы:

- Tiisto
- Ultra NatE
- P%oS

Они остались после попытки удалить родительский каталог. Пытаясь удалить их, я получаю такое сообщение:

```
rm: cannot remove 'ultravoxti': No such file or directory
```

Статистика о файле дает тот же результат. Команда ls -l показывает индексный дескриптор (inode), но если я пытаюсь схитрить командой

```
find -inum <index> -exec rm '{}' \;
```

то пропадает весь каталог:

```
find: ě./ultravoxti: No such file or directory
```

Я попробовал использовать символы шаблона, но проблема осталась. Удалить файлы через GUI я тоже не могу — в Thunar они не отображаются. Проверка диска с помощью fsck проблем не выявила.

spaceyhase

Как Вы запускали fsck? Если Вы делаете это на файловой системе ext2/3/4 без аргументов, то команда проверит, был ли диск корректно отмонтирован, сколько раз он был примонтирован и не превышает ли время, истекшее с момента последнего запуска fsck, установленного предела. Если ни одно из перечисленных условий не нарушено, то дальнейших проверок не выполняется. Для гарантии, что fsck действительно проверит всю файловую систему, необходимо добавить в эту команду опцию -f (force):

`fsck -f /dev/sdX`

и запустить ее при отмонтированной файловой системе. Если у Вас отдельный раздел `/home`, можно сделать это, выйдя из среды рабочего стола, войдя в консоль как `root` и отмонтировав `/home` перед запуском `fsck`: но простейшим, а в корневой файловой системе и единственным, вариантом будет загрузка с Live CD с целью запуска `fsck`.

Если файловая система не видит файлы, это может быть обусловлено тем, что имена файлов содержат символы, не распознаваемые системой. Первое, что я попытался бы сделать – воспользоваться автодополнением по нажатию клавиши Tab, потому что это, как минимум, позволит убедиться в том, что спецсимволы корректно экранируются. Следующий шаг – переименовать файлы так, чтобы их имена содержали только стандартные символы ASCII, и удалить их. Я также попытался бы переместить их в каталог, а затем удалить весь каталог командой `rm -fr`, но это Вы уже делали.

Если ничто не помогает, примените метод «грубой силы» – скоман্ডуйте

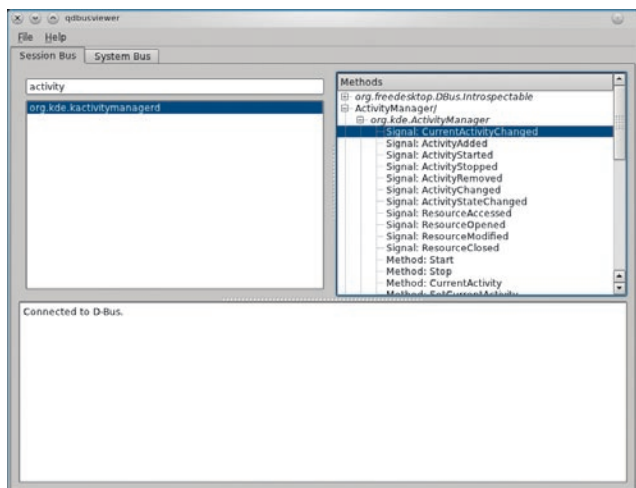
```
rm -i *
```

и непрерывно нажимайте клавишу `n` до тех пор, пока не дойдете до файла с некорректным именем. Это уже предпоследнее усилие – если и это не удастся, то Вам останется только сделать резервную копию, переформатировать раздел и провести восстановление; но будем надеяться на лучшее. **ПХ**

5 Смешение языков

В Я пытаюсь изучать русский язык и создал комнату [Activity], настроенную на помощь соответствующими плазмоидами и ссылками. Но я всегда считал, что глубокое погружение в язык помогает быстрее его осваивать; нет ли способа изменить язык дистрибутива в отдельной комнате? Как минимум – можно ли, войдя в эту комнату, автоматически переключать клавиатуру, а затем, выходя из нее, возвращаться к английскому языку? У меня OpenSUSE 11.4 с KDE 4.6.

Джеймс Даффи [James Duffy]



➤ Запутались в командах D-BUS? Гляньте, что вам доступно из `Qdbusviewer`.

Установить другой язык или раскладку клавиатуры для комнаты напрямую нельзя, и пока что нет опции автозапуска команды при смене комнаты; но это не значит, что Вы просите о невозможном. Вам надо копнуть чуть глубже – в частности, изучить систему D-BUS, которую KDE и другие программы используют для общения. D-BUS может и получать сигналы от KDE, и отправлять ему команды. Следующий скрипт слушает сигналы `CurrentActivityChanged` и выполняет действия, соответствующие текущей комнате.

```
#!/bin/sh
dbus-monitor "type='signal',path='/ActivityManager',member='CurrentActivityChanged'" \
| awk -F" " '{string="[0-9a-f]"; print $2; fflush(); }' |
while read ID; do
    NAME=$(qdbus org.kde.kactivitymanager /ActivityManager org.kde.ActivityManager.ActivityName $ID)
    if [ "$NAME" == "Russian" ]; then
        qdbus org.kde.keyboard /kxkb org.kde.KXKB.setLayout ru
    else
        sed -i -e 's/^Language=.*Language=ru/' -e 's/^Country=.*Country=ru/' ~/.kde4/share/config/kdeglobals
    fi
done
```

Код выглядит сложным, но в основном из-за имен, используемых D-BUS. Первая строка запускает команду `dbus-monitor`, прослушивающую сигналы `CurrentActivityChanged`. Команда `awk` извлекает идентификатор комнаты [activity's ID] из этого сообщения и возвращает запрос `qdbus`, чтобы получить название этой комнаты. Затем принимает простое решение `if...else`, на основе имени комнаты. Если это "Russian", то другая команда D-BUS переключает раскладку клавиатуры, а команда `sed` изменяет настройки страны и языка в файле

`kdeglobals` (установка локальных стандартов в KDE с помощью D-BUS пока невозможна). Запустите этот скрипт из модуля KDE Autostart, чтобы он ждал своего часа для переключения между комнатами.

Приведенный скрипт делает именно то, о чем Вы спрашивали, если это возможно. Здесь есть одна проблема, и, возможно, из-за нее пока и нет команды D-BUS для переключения локализации.

Переключение локали не влияет на уже работающие программы, а также на программы, которые

были недавно запущены и закрыты и оставили за собой в памяти код библиотек. Поэтому после переключения Вы получите смесь двух языков. При смене локали в KDE System Settings Вы получите соответствующее предупреждение – это особенность системы, и ее нельзя обойти.

Удачи Вам в изучении русского языка – я сам пытался учить его в школе, но надеюсь, Вы добьетесь лучших успехов, чем я. **НБ**

6 Автоматическая «расстройка»

В Я хочу сделать мой устаревший ноутбук Sony PCGFX109K (Pentium III, 256 МБ ОЗУ, диск 80 Гб, экран 1400×1050 SXGA+ TFT) полигоном для испытаний дистрибутивов Linux, поставляемых на DVD *Linux Format*. Но ни один из них не запускается корректно с Live CD на этой машине – на всех наблюдаются мерцание экрана, низкое разрешение или то, и другое.

Purpy 5.11 – единственный дистрибутив Linux, запустившийся с правильным разрешением и частотой обновления. Остальные – и большие, и маленькие – по умолчанию берут разрешение 800×600 и предоставляют лишь ограниченный диапазон настроек экрана и монитора.

Я пробовал различные варианты `vga=` при загрузке, но так и не сумел обойти ошибку, из-за которой Linux не распознает требования к экрану. Попытался редактировать файлы `/etc/X11/xorg.conf`, используя настройки из моего файла `xorg.conf` из Purpy, но в более современных дистрибутивах такого файла нет.

Где все эти дистрибутивы хранят экранные разрешения? Как принудительно установить в Linux разрешение 1400×1050 и соответствующую частоту обновления? Нет ли общего решения, пригодного для всех дистрибутивов?

Билл Флеминг [Bill Fleming]

Параметры `vga` (и `vesa`), передаваемые ядру при загрузке, управляют разрешением для текстовой консоли и не влияют на X.

По части файла `xorg.conf` Вы пошли верным путем. Причина, по которой в большинстве дистрибутивов сейчас нет такого файла, заключается в том, что в большинстве случаев X может определять графический адаптер, дисплей и устройства ввода, и настраивает все автоматически.

Это исключает потребность в создании файлов `xorg.conf` при установке, не говоря уже о проблемах, возникающих при замене оборудования, когда файл `xorg.conf` больше не годится.

Но X все-таки использует этот файл, если он есть. Настройки в этом файле приоритетнее, чем определенные автоматически, так что можно включить разделы для настроек монитора и экранного разрешения из имеющегося у Вас рабочего файла `xorg.conf` и предоставить системе автоматически определять все остальное, а то и вообще определить в этом файле все настройки.

Воспользуйтесь файлом `xorg.conf` из другого дистрибутива или создайте новый, запустив терминал и скоман্ডовав

```
sudo /etc/init.d/gdm stop
```

Эта команда закроет X. Затем нажмите клавиши Alt+F1, чтобы зарегистрироваться с консоли. Команда приведена для Ubuntu и в других дистрибутивах может меняться (некоторые дистрибутивы используют `/etc/init.d/dm`, `/etc/init.d/kdm` или `/etc/init.d/xdm`). Зайдите в консоль и дайте команду

```
sudo X -configure
```

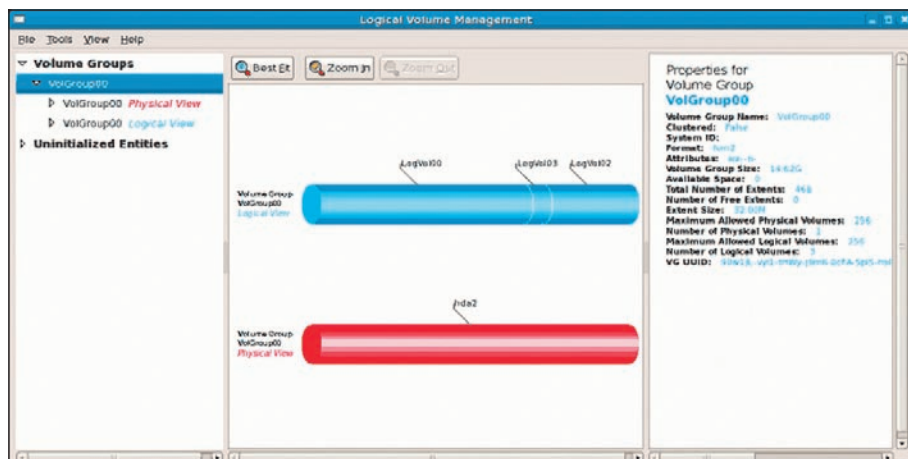
Она настроит X и запишет конфигурацию в файл `xorg.conf.new` в Вашем домашнем каталоге. Внесите нужные модификации и скопируйте файл в `/etc/X11/xorg.conf`. Перезапустите X указанной ранее командой, но вместо опции `stop` используйте опцию `start`, и X запустится с правильным разрешением.

Между прочим, хотя X по-прежнему использует `/etc/X11/xorg.conf`, идет переход на использование множества файлов из каталога `/etc/X11/xorg.conf.d`. Синтаксис идентичен, но если у Вас отдельные файлы для экрана, монитора, устройств ввода и т.д., это упрощает настройку.

Способ выбирайте сами, потому что поддержка старого монолитного файла еще какое-то время сохранится. **FM**

7 Утраченный том

В Я работаю с Fedora 12 и собираюсь обновиться до Fedora 14, предварительно наведя порядок в своих настройках LVM. Сейчас у меня четыре идентичных диска по 250 Гб каждый: `sda`, `sdb`, `sdc` и `sdd`. Устройство `/dev/sda` используется для Windows XP, и о нем речи нет. У меня есть две группы томов: `vg_yellow` и `vg_yellow12`.



Графическая программа настройки LVM в Fedora удобна, но истинную мощь LVM можно почувствовать только в командной строке.

Группа томов `vg_yellow12` включает только `sdd2`, а `sdd1` — загрузочный раздел. На этой группе томов установлена моя текущая система Fedora 12. В другой группе томов, `vg_yellow`, содержатся `sdc2`, `sdb1` и неизвестный [unknown] раздел, а `sdc1` — это ассоциированный загрузочный раздел. Эта система использовалась для более старой версии Fedora.

Мне бы хотелось удалить неизвестный раздел из группы `vg_yellow` или переименовать его — я думаю, что это часть `sdb`, и, наверное, переименовать его нужно в `sdb2`. LVM не позволяет мне удалить его, заявляя, что для его содержания недостаточно места. На графическом изо-

бражении видно, что в нем 59 618 экстендов, так что, предположительно, его размер — такой же, как у других двух разделов, а это не может быть правдой.

Коллин [Colin]

Информация из `pvs` и `vgs`, которую Вы предоставили, показывает:

```
Couldn't find device with uuid
210bCR-8zEM-8e2p-6xz3-k916-Xv8G-rf0I3e.
PV VG Fmt Attr PSize PFree
/dev/sdb1 vg_yellow lvm2 a- 232.88g 0
/dev/sdc2 vg_yellow lvm2 a- 232.69g 0
/dev/sdd2 vg_yellow12 lvm2 a- 232.69g 0
```



Часто задаваемые вопросы

Чтение файловых систем Windows

Можно ли монтировать диски Windows в Linux?

Да, можно. Более старые файловые системы MS-DOS — FAT и VFAT — поддерживаются уже много лет: большинство установщиков дистрибутивов автоматически обнаруживают файловые системы FAT и VFAT и монтируют их. Для ручной монтирования файловой системы командуйте

```
mount -t vfat /dev/hda1 /mnt/windows
```

А как насчет файловой системы NTFS из XP?

До недавнего времени в большинстве дистрибутивов поддержка NTFS была неполной: в лучшем случае предлагался доступ с правом чтения. Ядро Linux по-прежнему воспринимает запись как экспериментальную опцию, которая часто бывает заблокирована, но те-

перь имеется драйвер FUSE, под названием `ntfs3g`, предоставляющий доступ к файловым системам NTFS с правом чтения и записи; ищите его на <http://www.tuxera.com/community/ntfs-3g-download/>. Драйверы Fuse работают не так, как файловые системы, реализованные на уровне ядра. Раздел NTFS монтируется командой:

```
ntfs-3g /dev/hda1 /mnt/windows
```

а для отмонтирования служит команда

```
fusermount -u /dev/hda1
```

Я добавил мой раздел Windows к /etc/fstab. Почему туда может писать только пользователь root?

В FAT нет концепции владельца или прав доступа, и владельцем всех файлов и каталогов становится пользователь, примонтировавший файловую систему, а во время за-

грузки это делается от имени `root`. Изменить владельца можно, указав опцию `uid` в строке `fstab`, но полный доступ все равно получит только один пользователь. Альтернатива — изменить настройку `umask`, управляющую правами доступа к файлам. Следующая строка в `/etc/fstab` разрешает доступ к файлам всем пользователям, а принадлежат эти файлы пользователю `fred` из группы `users`.

```
/dev/hda1 /mnt/windows vfat
umask=000,uid=fred,gid=users 0 0
```

Чем различаются FAT, VFAT и FAT32?

FAT — это изначальная файловая система MS-DOS, а VFAT — ее расширение для Windows 95, умеющее обрабатывать длинные имена файлов. FAT32 — дальнейшее усовершенствование, для работы с боль-

шими жесткими дисками. Разделы жестких дисков практически всегда формируются под FAT32, а флэш-накопители — обычно под FAT16. Вам незачем об этом беспокоиться, кроме как при форматировании устройства.

Читаются ли файловые системы Linux из-под Windows?

Существует драйвер для `ext2/3`, доступный по адресу www.fs-driver.org; он монтирует файловую систему `ext2` как отдельный диск с собственным буквенным обозначением. Он не полностью интерпретирует права доступа, потому что установленная копия Windows ничего не знает о пользователях Linux. Кроме того, существует утилита `rfstool` (<http://p-nand-q.com/download/rfstool.html>) для доступа к разделам ReiserFS из Windows.

unknown device vg_yellow lvm2 a- 232.88g 0

Обратите внимание на размер неизвестного устройства – полных 250 ГБ, ровно столько же, сколько и устройство /dev/sdb1. Должно быть, раньше Вы либо подключали к компьютеру еще одно устройство (возможно, внешний жесткий диск) или настроили один и тот же раздел как физический том более одного раза. Подозреваю, что Вы добавили sdb1 во время предшествующей установки, а затем выбрали его еще раз во время установки Fedora 12. Поскольку для идентификации разделов LVM использует UUID, создаваемый для устройства в момент его инициализации, а не узлы устройств, которые могут измениться, оригиналь-

ное воплощение физического тома теперь пропало. К сожалению, пропали и данные, которые могли на нем храниться.

Попробуйте удалить отсутствующий неизвестный том следующей командой:

```
vgreduce --removemissing vg_yellow
```

Команда должна работать, пока LVM не попытается использовать данные на отсутствующем томе. Если это так – а я подозреваю, что это так – добавьте опцию --force и распрощайтесь с находившейся там информацией.

```
vgreduce --removemissing --force vg_yellow
```

Это весьма разрушительная команда – она удаляет все логические тома вместе с их содержимым,

даже если они лишь частично хранились на отсутствующем устройстве; но уж раз Вы дошли до этой стадии, то, скорее всего, установленная копия серьезно повреждена, и Вам остается только, стиснув зубы, сделать это.

Сообщение о недостатке свободного пространства появляется потому, что Вам потребовались свободные экстенды (в LVM это аналог блоков дискового пространства); откуда следует, что они не выделены ни одному из логических томов, и свободное пространство на существующих томах Вам помочь не в состоянии. Вывод vgs не показывает свободных экстендов ни в одной из групп томов.

Терминалы и суперпользователи

Мы часто предлагаем решение в виде команд, вводимых в терминале. Хотя обычно то же самое можно выполнить и через графические инструменты, входящие в состав дистрибутива, отличия между этими инструментами означают, что такие решения будут узко специфичными. Команды терминала гораздо более гибкие и работают во всех дистрибутивах.

Команды настройки системы часто должны запускаться от имени суперпользователя (superuser или root). В зависимости от вашего дистрибутива, есть два способа сделать это. Многие дистрибутивы – в частно-

сти, Ubuntu и его производные – требуют предварять такие команды префиксом sudo. После этого пользователю предлагается ввести пароль, и система выдает ему полномочия root, действующие на время выполнения команды. Другие дистрибутивы используют префикс su, применение которого требует ввода пароля root и дает полные права root до тех пор, пока пользователь не введет команду logout. Если ваш дистрибутив использует su, выполните эту команду один раз, и потом сможете выполнять любую команду, не предваряя ее префиксом sudo.



Краткая справка про...

Ссылки на файлы

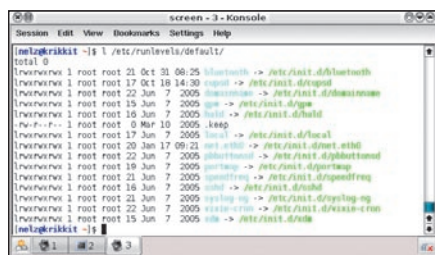
Linux активно использует ссылки на файлы, в основном символические (или мягкие – soft). Ссылка на файл – это способ держать один и тот же файл (или каталог) в двух или более местах или под различными именами. Файл можно просто скопировать, но это – транжирство дисковой памяти; кроме того, при обновлении исходного файла копия остается старой.

Файловые системы Linux предоставляют средства доступа к одному и тому же файлу или каталогу из нескольких местоположений – так называемые ссылки. Существуют два типа ссылок: жесткие [hard] и символические [symlinks]. Технически, любой файл – это жесткая ссылка, связывающая имя файла со структурой данных, где хранится его содержимое. Говоря о жестких ссылках, мы имеем в виду дополнительные ссылки, за счет которых один файл может иметь несколько имен. Жесткие ссылки имеют ограничения; самое важное из них – то, что они применимы только к файлам, но не к каталогам, и то, что все ссылки должны находиться в одной и той же файловой системе.

Символические ссылки гораздо более гибкие, и в типичной системе Linux вы встретите

их повсюду. Для их просмотра используйте опцию -l с командой ls, и они отображаются большинством файловых менеджеров. Классический пример можно найти в каталоге /usr/src, где /usr/src/linux – символическая ссылка на текущее ядро, скажем, Linux 2.6.15. За счет этого можно установить исходные коды нескольких ядер, и любая программа, требующая доступа к исходному коду ядра, ищет необходимый ей исходный код в /usr/src/linux. Множество символических ссылок имеется и в /etc/rc.d.

Символические ссылки можно создать командой ln -s /path/to/file /path/to/link. И исходный, и целевой пути могут быть относительными.



➤ Масса символических ссылок находится в каталоге /etc, где файлы должны быть доступны.

8 Где Windows?

В Я воспользовался инструкциями, приведенными в LXF143, для создания системы с двойной загрузкой Ubuntu и Fedora, с небольшой модификацией. У меня старый компьютер и слишком малый объем памяти, чтобы ставить рискованные эксперименты – и вместо виртуальной машины я воспользовался вторым жестким диском, абсолютно пустым, а на первом диске была Windows.

Я успешно установил Fedora и Ubuntu, и после перезагрузки с удовлетворением обнаружил загрузочное меню. Но в нем, к сожалению, не было Windows. Запустив sudo update-grub, я получил сообщение об ошибке:

```
“Cannot access /var/lib/os-prober/mount/boot
Boot: No such file or directory”
```

Тогда я заглянул в /var/lib/os-prober и нашел там файл labels – но никакого следа /mount/boot. Как решить эту проблему? Команда sudo fdisk -l показывает, что Windows загружается с /dev/sda1, а Fedora и Ubuntu – с /dev/sdb1.

Крис Крофтон-Слей [Chris Crofton-Sleigh]

Такое бывает, когда пытаются установить Grub2 с разделом Windows, выбранным в качестве корневого каталога.

В результате каталог /boot/grub создается на разделе Windows; но если там уже есть каталог /Boot, возникает путаница – файловые системы Linux интерпретируют boot и Boot как разные каталоги, а файловые системы Windows – нет. Решение – загрузить Ubuntu и примонтировать раздел Windows. Ubuntu должна монтировать его автоматически, а если не монтирует, можно сделать это вручную из терминала командами:

```
mkdir -p /mnt/windows
sudo ntfs-3g /dev/sda1 /mnt/windows
```

Затем удалите каталог boot из /mnt/windows или того каталога, где он монтирован, предварительно убедившись, что удаляете именно boot, а не Boot, так как последний нужен для загрузки Windows. Теперь скомандуйте

```
sudo os-prober
```

Если эта команда обнаружит установленную копию Windows, скомандуйте

```
sudo update-grub
```

чтобы добавить Windows в меню Grub. Перезагрузитесь, и Вы увидите в загрузочном меню опции для Fedora, Ubuntu и Windows. MC

Большой вопрос Каков лучший способ администрирования серверов?

В Я ищу наилучший серверный GUI для Linux, позволяющий управлять доменными пользователями и группами, общим доступом к файлам и т.д. Поискал в distrowatch.com/search.php?category=Server, но не нашел. Я говорю о GUI, потому что по работе я знаком с Windows Server 2008 и хотел бы попробовать дома аналог для Linux.

Роб Фримен [Rob Freeman]

Серверы и GUI особо не дружат. Сервер работает в фоновом режиме, и рабочий стол ему не требуется. У большинства серверов даже нет монитора, так что рабочий стол для них не только бессмыслен, но и зря потребляет системные ресурсы. Управляются сервера обычно удаленно – либо через удаленную консоль по SSH, либо через web-интерфейс. Самый известный из них – *Webmin*, входящий в репозитории большинства дистрибутивов; его можно установить обычным образом, а потом получить к нему доступ через браузер, введя адрес <https://localhost:10000>.

Первой остановкой будет IP Access Control в разделе Webmin Configuration. Задайте список

IP-адресов, с которых разрешен доступ, чтобы Вы и только Вы могли регистрироваться с других компьютеров. Затем просмотрите опции настройки разных типов сервера – большинство из них сидит в разделе *Samba*: именно Samba управляет общим доступом к файлам Windows и ассоциированными сервисами. Добившись работы *Webmin* с другого компьютера, можно перейти в раздел *Services* и убрать запуск графического интерфейса после перезагрузки.

Если Вы хотите просто попробовать на своем компьютере всякие серверы, серверный дистрибутив вовсе не нужен. Популярное серверное ПО входит в состав всех дистрибутивов. Например, ради экспериментов с общим доступом к файлам Windows установите себе *Samba*.

Если Вы хотите испробовать специализированный серверный дистрибутив, хорошее начало – Ubuntu Server. В его состав входит полный набор серверного ПО; выберите при установке то, что Вам требуется в первую очередь, а впоследствии добавьте остальные по мере необходимости. Имеющуюся утилиту настройки на базе Web нужно устанавливать отдельно. Текстовый установщик этого дистрибутива может пока-

заться странным, но делает он то же самое. После установки перезагрузите компьютер и войдите как пользователь, созданный в процессе установки. Затем установите *eBox* командой:

```
sudo apt-get install ebox
```

Возможно, Вы захотите изменить порт, используемый во время установки. По умолчанию это порт 443, стандарт для HTTPS. Но если Вы потом захотите завести web-сервер, этот номер может вызвать конфликт, и лучше изменить его на какой-нибудь неиспользуемый номер – например, 441. Сделать это можно и через web-интерфейс.

Завершив установку, введите в web-браузере адрес <https://server.address:441>, чтобы увидеть интерфейс. Он скудноват, и Вам потребуются установить модули для сервисов, которыми Вы хотите управлять. Вернитесь в терминал и скомандуйте

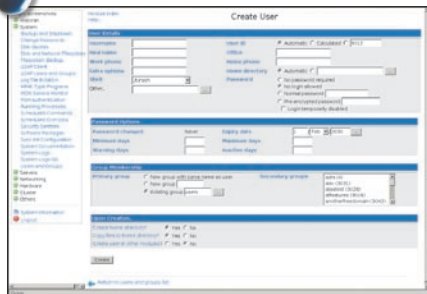
```
apt-cache search eboxto
```

Вы увидите список доступных модулей. Затем установите нужные Вам модули, например:

```
apt-get install ebox-samba eboxusersandgroups ebox-printers
```

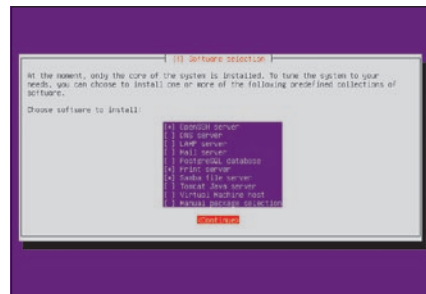
Теперь Вы сможете выполнять свои задачи через web-браузер. **НБ** **LXF**

Шаг за шагом: Настроим web-интерфейс для сервера



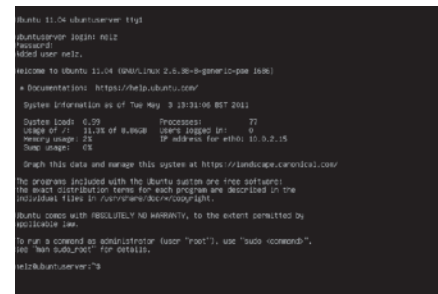
1 Webmin

Webmin – это GUI для администрирования через браузер, применимый в большинстве дистрибутивов для настройки и поддержки широкого диапазона ПО.



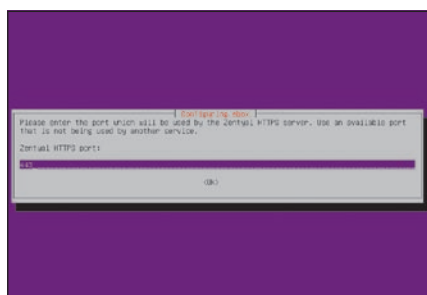
2 Установка Ubuntu Server

GUI установщика выглядит примитивным, но работает не хуже любого другого. Здесь выбираются типы требуемого к установке серверного ПО.



3 Смотреть особо не на что...

Такую картину вы увидите по завершении установки Ubuntu Server. Внешняя красота здесь и не планировалась, ведь вся работа выполняется в фоновом режиме.



4 Установка eBox

Благодаря этому вы получите web-интерфейс администратора, но, возможно, захотите изменить порт по умолчанию, особенно если намерены держать web-сервер.



5 Еще немного команд

Последний раз вам понадобится командная строка, чтобы с помощью команд *apt-cache* и *apt-get* искать и устанавливать модули *eBox* для требуемых вам сервисов.



6 Вот и все

Web-клиент Zentyal, предоставляемый *eBox*, позволяет наблюдать, настраивать и поддерживать серверное ПО через web-браузер.

LXF HotPicks



Алекс Кокс
Составляя HotPicks, Алекс неистово ругался, аж до посинения волос.

Lyx Document Processor » SwarmTV » LIRC » mhWaveEdit » Weechat
» Rekonq » Conquests » Abuse » White Dune » Clonezilla

Программа для работы с документами

Lyx Document Processor

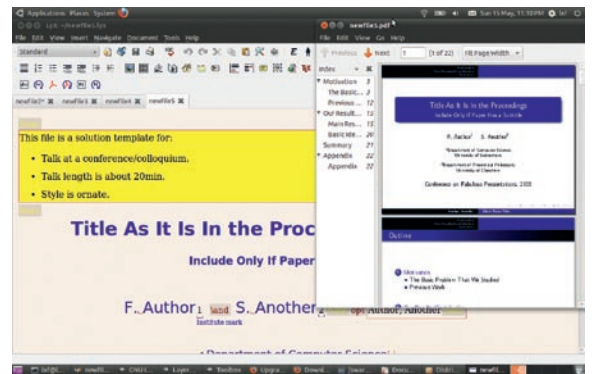
Сайт www.lyx.org

Текстовые редакторы – штука странная. Почему мы вообще чувствуем потребность обрабатывать текст? Не поместить ли эти пакеты в ту же категорию, что и, скажем, кухонные процессоры? Сунуть в него пригоршню глаголов и пучок существительных, sprysnut прилагательными, а потом, плюнув на все ингредиенты, выкручивать кнопки, пока не получится некая неудобоваримая смесь?

А вот *Lyx* так не считает. Видите ли, *Lyx* – вовсе не текстовый редактор. Это

«редактор документов». Он полагает, что мы слишком много возмисм с внешним видом наших документов, отвлекаясь от куда более важной задачи их написания. Вместо выдачи вам ленты со множеством опций форматирования, *Lyx* следует принципу WYSIWYM [What You See Is What You Mean] – то есть форматирование ваши документы в соответствии не с тем, что вам выдано, а с тем, что вы имеете в виду.

Вы, поди, уже смекнули по названию, что *Lyx* – это интерфейс для *LaTeX*. При создании документов в *Lyx* форматирование добавляется исходя из их типа, а затем документы скармливаются



» Преобразуйте стандартный документ в PDF и взгляните, как он смотрится в печатном виде. Разница огромна.

LaTeX для финального рендеринга и сохранения в выбранном формате. С учетом связи с *LaTeX*, не очень удивляет, что главная сила *Lyx* – в технической и научной документации. В нем есть превосходный инструмент написания математических формул на основе принципа наведи-и-щелкни [point-and-click]: он генерирует до того классный код, что и в чистом *LaTeX* заставил бы самого здравомыслящего ученого создать супермонстра, способного уничтожить весь мир. Если *LaTeX* вам не в диковину, можете отставить point-and-click и применять здесь те же самые кнопки быстрого запуска и коды.

Помимо своей научной библиотеки, *Lyx* скопил множество типов документов, чуть более привычных простым смертным и позволяющих создать все, что угодно: от писем и статей до книг и киносценариев. Будь у меня время, я бы не поленился сделать специальный тип для **LXFHotPicks**, но увы... придется Грэму мучиться с этим неформатируемым документом-катастрофой. Поплачь о нем, читатель.

Освоение *Lyx* требует времени – работа в *LaTeX* явно виртуознее, чем штамповка стандартных документов в *LibreOffice*; но ваши усилия не пропадут даром. Если ваша работа – создание документации, *Lyx* изменит ваше мнение о писательстве.

«Вместо внешнего вида *Lyx* использует принцип WYSIWYM.»

Пробираемся через *Lyx*

Выбор стиля [Style picker]

Это выпадающее окно задает стиль любому фрагменту текста, который вы в данный момент редактируете.

Рендеринг [Render up]

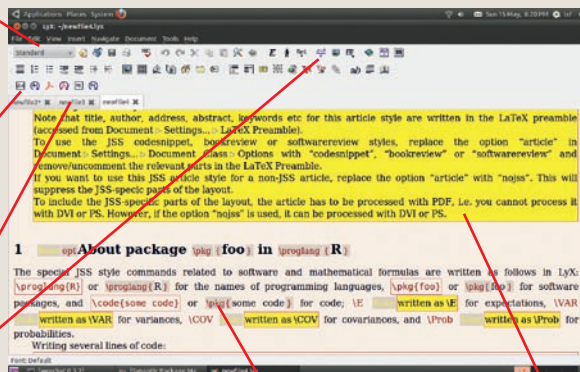
Текущий документ превращается в файл DVI, готовый для печати, одним нажатием этой кнопки.

Вкладки документов [Document tabs]

Можно одновременно редактировать, копировать и вставлять в несколько документов.

Добавление формул [Add maths]

Эта кнопка вызывает окно формул (оно появится в голубой рамке).



Встроенный код [Inline code]

Кнопка TeX во втором ряду инструментов позволяет встроить стандартный код *LaTeX* в ваши документы.

Примечания [Take note]

При помощи желтой кнопки вверху экрана вносятся примечания. Они не подвергаются рендерингу.

Инструмент поиска программ

SwarmTV

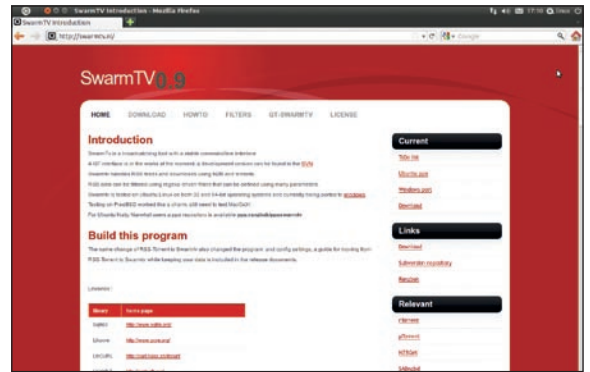
Версия 0.8 Сайт www.swarmtv.nl

Сперва – стандартное предупреждение: *Linux Format* не одобряет незаконное скачивание того, что вам не принадлежит, использование Интернета в неблагоприятных целях и ковыряние в носу сидя в туалете. Но мы вполне осознаем, что все три перечисленных действия доступны каждому. Получается турик? Тогда давайте ограничимся реформированием ТВ-программ, записи которых мешает только неудобный формат, и редкими приколами в Facebook – а с чистой совестью в кабинке это не к нам, мы далеко.

SwarmTV – довольно незатейливый демон командной строки, который внедряется в ленту RSS какого-либо торрента или к провайдеру NZB, отыскивает нужную вам программу с помощью настроенного пользователем фильтра и помещает торрент или файлы NZB в определенную вами папку.

Начните с добавления источника. Это делается в командной строке, и инструкции на сайте достаточно подробны; короче, вам надо использовать расширения `--add-source "NAME"` для определения вашего источника в базе данных *SwarmTV*, затем `--url="http://your.url"` для указания на ленту RSS, где вы ищите нужное, `--source-parser "type"` для определения вида ленты (как правило, `defaultrss`) и, наконец, `--metatype "type"` для сообщения *SwarmTV*, что вы хотите закачать: файлы NZB или файлы торрента. Учтите, что *SwarmTV* ничего не скачивает, а только находит нужный источник; вам придется настроить свои программы

«SwarmTV — довольно незатейливый демон командной строки.»



› Автоматическая загрузка – дело всего лишь поиска нужной ленты.

для загрузки (достойный выбор – *SabNZBd* или *rTorrent*) на слежение за папками, куда эта загрузка осуществляется.

Добавьте также фильтр для выбора того, что будет закачено из предварительно определенных источников. Используйте `--add-simple 'name'` для определения его внутреннего имени и `--title='^title'`, чтобы найти искомые серии – символ `^` означает «начать с». `--max-size` и `--minsize` задают ограничения по размеру, чтобы получить стандартное разрешение или версию HD. Загляните на map-страницу *SwarmTV* ради более подробной информации о простых фильтрах, и приступайте. Только ведите себя хорошо, ладно?

Дистанционное управление

LIRC

Версия 0.9.0 Сайт www.lirc.org

Современные дети, со всеми своими Bluetooth и Wi-Fi, не в курсе, в каком мире родились. Когда ребенком был я, для запуска ZX81 издали приходилось применять чуть ли не старомодную кочергу. Сейчас-то я понимаю, что это был садистский инструмент. Нажать им на мембранную клавишу без риска свернуть модуль ОЗУ было почти невозможно. Не знаю, зачем мы вообще это делали.

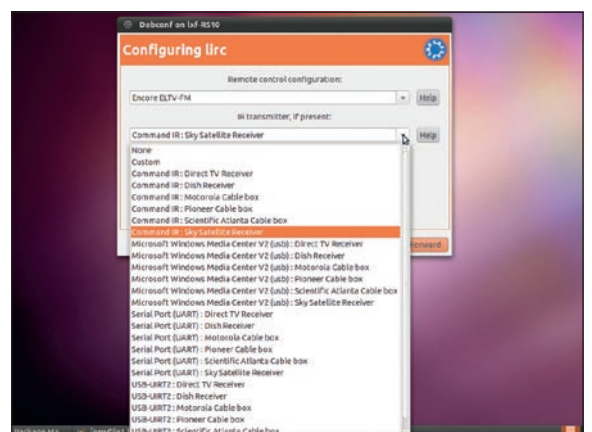
«Кочерга» не сдавала своих позиций в эру ST и Amiga, и в пору моих ранних ПК. Но когда ПК стали ставить под телевизор, пришло время освоить технику, перейдя на инфракрасное излучение. Что и породило *Linux Infrared Remote Control*: эх, мне бы этот инструмент, когда я настраивал древние медиа-центры!

В зависимости от вашего оборудования, *LIRC* работает и в одну, и в другую сторону. Он может действовать как ИК-передатчик, отправляя сообщения вашим ИК-совместимым устройствам, или

как приемник, превращающий практически любой ИК-сигнал во внутренние команды. Однако это не значит, что «play» на пульте превращается только в «play» внутри. Например, можно преобразовать клавиши со стрелками в перемещение курсора, превратив свой пульт в рудиментарную (и довольно неуклюжую) мышь.

Не стоит недооценивать и широко-вещательную ИК-передачу: настройте компьютер как удаленный повторитель, сделав сигналы с одного пульта принимаемыми множеством различных устройств. Вся ваша ИК-периферия вряд ли поддерживает один и тот же набор команд – тут понадобится немного кода; вы можете без проблем использовать пакет инструментов командной строки.

«Настройте компьютер как удаленный повторитель ИК-сигналов.»



› *LIRC* поддерживает множество приемников, от специальных до встроенных в ТВ-карты.

Желателен пример? Ну так найдите нужное оборудование в списке поддерживаемых устройств и воспользуйтесь `irsend SEND_ONCE DeviceName VOL_UP`, чтобы изменить громкость. Легко! И кому нужна кочерга, с такой-то техникой?

Звуковой редактор

mhWaveEdit

Версия 1.4.21 Сайт www.gna.org/projects/mhwaveedit

Будем проще (KISS!): перед вами – *mhWaveEdit*. Я расписал бы его – потерпите, это моя аналогия, чтоб превратить заурядную программу в нечто слегка волнующее – как программный эквивалент кассеты C90 в кассетнике.

К счастью, это не кассетник в сомнительной MIDI-системе 80-х, увековечивающей искаженные голоса и удваивающий скорость воспроизведения из-за того, что какой-то телемагнат посквалыжничал выложить наличные денежки за продукт достойного качества.

Это, скорее, особое устройство. Скажем, Wharfedale, прямо от Richer Sounds начала 90-х. Он немного отстает от времени – уже есть устройства получше, с более сложными функциями. Но, правильно построив все его провода, вы сможете подключить его практически ко всему, и лента

«Запишите аудио на стереотрек, сохраните как WAVE-файл... все!»

в нем не провиснет. Мир записи целиком и полностью у вас в руках.

Думаю, кассетных аналогий достаточно. Суть такова: *mhWaveEdit* не более чем деструктивный звуковой редактор. Запишите аудио на одиночный стереотрек, сохраните как WAVE-файл... все! Но если вашей системе добавить возможностей, ее ценность будет расти в геометрической прогрессии.

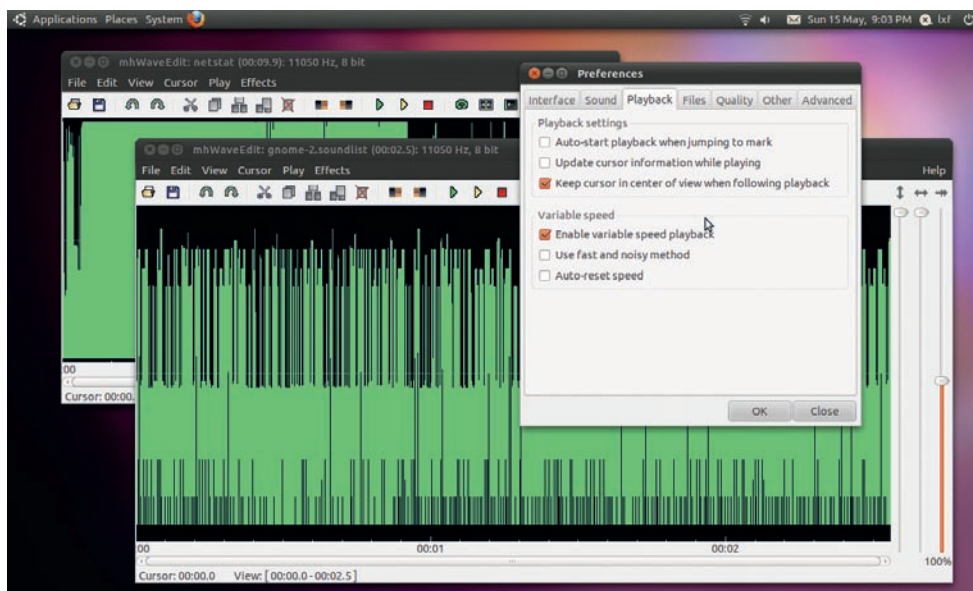
Если у вас установлен Jack, как оно и должно быть, вы можете подключать к *mhWaveEdit* все что угодно.

Используйте ALSA, и вы будете работать практически без времени задержки. Откопайте модули расширения LADSPA (на сайте www.ladspa.org), и возможностей станет больше прежнего. Или установите *Lame*, *oggenc* или *libsndfile*, чтобы осуществлять экспорт в другие форматы, помимо WAVE.

Стабильный и безопасный

mhWaveEdit лучше всего проявляет свои достоинства в долгосрочной записи; насколько мы можем судить, он очень стабилен, и поскольку пишет прямо на диск, готовые данные не пострадают, даже если вся ваша система выключится.

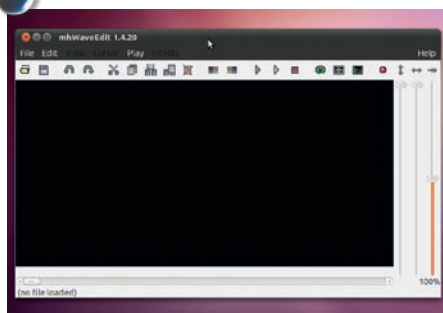
Будьте осторожны при вырезании и вставке частей; убедитесь, что работаете с копией оригинала, если не хотите случайно потерять данные, потому что *mhWaveEdit* не сохраняет истории для отката, в отличие от *Audacity*.



➤ Используйте панель Preferences с умом – она содержит весьма полезные дополнения.

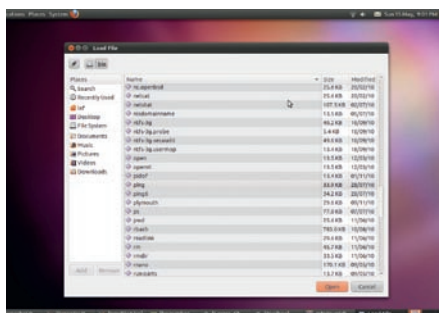


Шаг за шагом: Наука звука



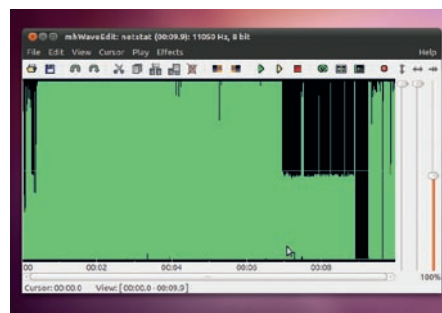
1 Загрузим...

mhWaveEdit довольно радушен. Он загрузит любой указанный файл, как если бы это был файл WAVE.



2 Откроем...

Попробуйте – используйте File > Open, и найдите, скажем, двоичный файл. Они, как правило, более упорядоченные, чем картинки.



3 Воспроизведем...

Воспроизведите его, и вы услышите звук начала 80-х, словно ZX Spectrum не умирал.

Клиент чата

WeeChat

Версия 0.3.5 Сайт www.weechat.org

IRC чреват опасностями. IRC видит, как вы, 17 лет от роду, садитесь на автобус до Лестера для встречи с друзьями по каналу. IRC видит, как вы встречаетесь со слепым мужчиной-микроцефалом, категорически асоциальной четырнадцатилеткой или замужней матерью троих детей, которая норовит вас соблазнить, пока ее вторая половина валяется в четырех футах от вас, упившись пивом. IRC, люди, ужасная вещь. Уберите его от меня.

Если же IRC еще не дал вам урок, уж лучше будет познакомиться с *WeeChat*, аккуратным расширяемым клиентом, который почти гарантированно, ни секунды не колеблясь, отправит вас по кривой дорожке интернет-общения тинейджеров.

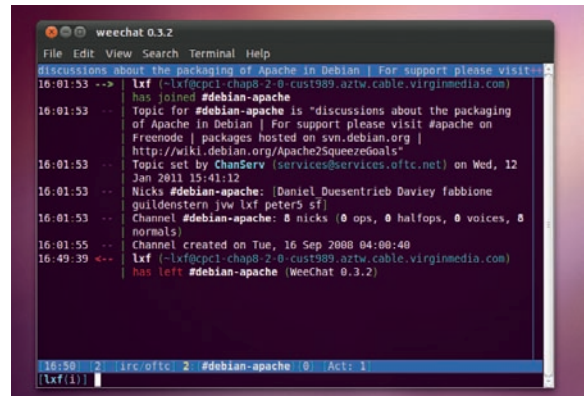
Он целиком и полностью текстовый — ядро в пакете *weechat*, но вам также придется установить *weechat-curses*, чтобы добиться от него чего-нибудь — так что запустите его из оболочки. Через модули расширения он поддерживает языки скриптов, типа Perl, Python и Ruby, и может

стать отличной платформой для управления IRC-ботом или иными созданиями.

Для начала надо знать несколько простых команд. Добавьте свой первый сервер с помощью `/server add <name> <address>`, указав на онлайн-сервер IRC и включив его порт после завершающего слэша. Например, сервер OFTC, работающий на порте 6667, будет `irc.oftc.net/6667`. Затем используйте `/connect <name>`, чтобы зайти на сервер, и примените стандартные команды IRC (`/list`, `/join`, `/nick` и прочие в том же роде), как только подцепитесь.

Учитывая способность *WeeChat* работать со скриптами, вы не удивитесь, обнаружив, что в нем уже имеется солидный выбор таковых для настройки его работы. Полный их список есть на www.weechat.org.

«IRC ВИДИТ, КАК ЗАМУЖНЯЯ МАТЬ ТРОИХ НОРОВИТ ВАС СОБЛАЗНИТЬ.»



➤ К *WeeChat* надо попривыкнуть, но встроенные устройства мощи — непревзойденные.

[org/scripts/](#), и здесь вы найдете все, от перекраски текста до самого возмутительного из всех скриптов IRC, интеграции медиа-плеера.

Хотите, чтобы кто-то из наугад выбранного канала узнал, что вы слушаете нарезку Marky Mark и Funky Bunch [популярная в начале 90-х гг. хип-хоп группа, — *прим. пер.*] на *Amarok*? Отчего бы и нет.

Браузер

Rekonq

Версия 0.7.0 Сайт <http://rekonq.kde.org/>

Новые браузеры не имеют привычки возникать на каждом шагу, и причина этому, вероятно, достаточно солидная. *Firefox* и *Chrome/Chromium* распилили рынок между собой, и по большей части они неплохо выполняют свою работу.

Штуки вроде *Flock*, ответвления кода с непомерными амбициями, не особо преуспевают у своей целевой аудитории, как видно по закрытию самого *Flock*. А если вы вознамерились повыпендриваться, вы скорее возьмете *Opera*, а не что-то мелкое и малоизвестное, верно?

Возможно, и нет. *Firefox 4* заставил наморщиться не один лоб из-за своего управления памятью, которое иначе как «ужасным» не назовешь. Он запарол больше систем, чем какой-нибудь Майк Сондерс (имя подлежит замене на ваше усмотрение). *Chrome* тоже не лишен проблем при работе с большим количеством вкладок, потому что резервирует по сегменту

ОЗУ для каждой. *Konquerer*? Да, есть такой. Причем огромный. Это, возможно, самый расширяемый браузер из всех ныне существующих. И для некоторых это серьезно — настолько, что в качестве ведущего браузера Kubuntu ему предпочитают *Rekonq*.

Rekonq (смеха ради, я решил произносить это как «Re-sonk [дать по носу]») затеял решить названные проблемы. Это браузер на основе KDE, работающий в среде *WebKit* (вместо выбранного в *Konquerer* по умолчанию *KHTML*), и понятно, что у него высокая совместимость с большим количеством страниц и обновляемость. Его философия — «маленький и легкий», и он усиленно тщится не быть пожирате-

«Его философия — “маленький и легкий”, и он не пожирает память.»



➤ *Rekonq* быстрый, легковесный и отлично разработан. Возможно, это сейчас лучший браузер в KDE.

лем памяти. В нем есть режим по умолчанию, позволяющий выбирать, когда загружать модули расширения, и подобный FlashBlock для *Firefox*, но с более широким доступом — замечательно, особенно когда надо сократить лишнее потребление памяти, а то и для экономии расхода батарейки при работе на ноутбуке.

HotGames Развлекательные приложения

Пошаговая стратегия

Conquests

Версия 0.12 Сайт <http://homepage.ntlworld.com/mark.harman/conquests.html>

Было бы справедливо сказать, что мы в сообществе Linux не чахнем от недостатка клонов *Civilization*. *FreeCiv* – возможно, самое полное воссоздание исходной *Civ* – шло в комплекте с большим числом дистрибутивов, чем у Майка Сондерса было снов, что он Сид Мейер [Sid Meier – разработчик игр, в т.ч. *Civilization*, – прим. ред.]. *FreeCol* сделала то же самое, но уже для *Colonization*. А теперь подняли головы *AltaCiv* и *Civquest*, и они жаждут внимания.

Мы, пожалуй, покончили с клонами *Civ* на всю оставшуюся жизнь. Нечего тратить на них силы.

Ой. Я отвлекся. *Conquests*. Извините.

Да, это клон *Civ*. Первый намек – помимо хрестоматийной формулы «построй город, создай ресурсы, расширь свою империю» – возникает в виде графики, взятой прямо из *FreeCiv*. Мудрое использование источников с откры-

тым кодом, скажем мы. Зачем искать добра от добра, если уже готовы вполне достойные текстуры? А вот в самой работе *Conquests* поднимает планку; здесь имеется сходный 3D-движок, и хотя такая фантастика, как колеблющиеся миры более поздних версий *Civ*, здесь отсутствует, базовая двумерная простота *FreeCiv* определенно украшена – например, игре придает прелести смешение текстов.

Игрокам в любую из *Civ* справиться с *Conquests* будет относительно просто. Здорово я вас удивил, а? Насчет подсказок – ищите бонусы: вы будете их получать за строительство городов на определенных территориях. Например,

«В зонах виноделия развита торговля, и города растут быстрее.»



» Переговоры с французами? Ни за что! Пока они не предложат стереотипно низкие пошлины на... лук. Да, на лук.

красным цветом выделены зоны виноделия, где развита торговля и города растут быстрее. Открыв силу пара, ищите регионы, отмеченные черным – уголь весьма ценен, поскольку значительно повышает производительность.

Есть и несколько загвоздок – самая неприятная в том, что нельзя использовать Alt+Tab, чтобы вернуться на рабочий стол Ubuntu; но будьте снисходительны к *Conquests*. Это всего лишь производное, которое пробивает себе дорогу на переполненном рынке; но игра действительно хороша.

2D side-scrolling platform

Abuse

Версия 0.8 Сайт <http://abuse.zoy.org>

Как ни приятно сидеть здесь, злоупотребляя [abuse] шутками про Майка, на сей раз я не поддаюсь соблазну и займусь делом. Вы когда-нибудь играли в смертельную схватку *Soldat* от Polish 2D или его духовного предшественника *Liero*? Если нет, то зря: обе великолепны. И обе многим обязаны *Abuse*. Это был прототип платформеров с боковой прокруткой, управляемых мышью; и вот сейчас, через 16 лет после выхода исходного релиза, у них открытый код и свободная графика. Чудесно.

Однако – не без оговорок. *Abuse* имеет открытый код и распространяется под GPL, но имели место проблемы с официальным лицензированием наследия его оригинала. Поэтому мы предоставляем вам на диске полностью свободный, но слегка урезанный tar-архив;

в нем сохранены все уровни и игровой сюжет оригинала, но для получения полной версии нужно посетить сайт или установить *Abuse* из менеджера пакетов вашего дистрибутива. В этом пункте он считается неконтролируемым – не в последнюю очередь из-за того, что уже не отследить, кто на самом деле владеет правами – и мы поместили его в ту же категорию, что и *OpenTTD*. Технически это не вполне законно, но это не незаконная версия, и никто на вас не пожалуется.

Запустите ее, и вы займете свое праведное место в малость запачканной б/у

«Прототип платформеров с боковой прокруткой мышью.»



» Перебежка. Выстрел. Увернуться от взрыва. Не вышло. Мучительная смерть. К повтору.

броню некоего Ника Вренна [Nick Vrenna], борца сопротивления, сражающегося с армией жутких генно-модифицированных Муравьев, которая настолько близка к киношным Пришельцам, насколько компания-производитель Crack Dot Com смогла к ним подобраться, не рискуя обременить свой почтовый ящик обвинением на 1000 страниц. И хотя решение головоломок тоже понадобится, главной в повестке дня остается рефлексорная пальба. Заболтать Муравьев не удастся, но вы определенно сможете пустить им пулю в лоб.

Инструмент создания 3D

White Dune

Версия 0.31beta211 Сайт <http://vrml.cip.ica.uni-stuttgart.de/dune/>

Автор *White Dune* прислал нам любезное личное письмо и сообщил, что с момента последнего обзора программы в LXF101 она претерпела ряд изменений. Честолюбивым разработчикам на заметку: весьма недурно так поступать как раз накануне составления HotPicks. Кхм.

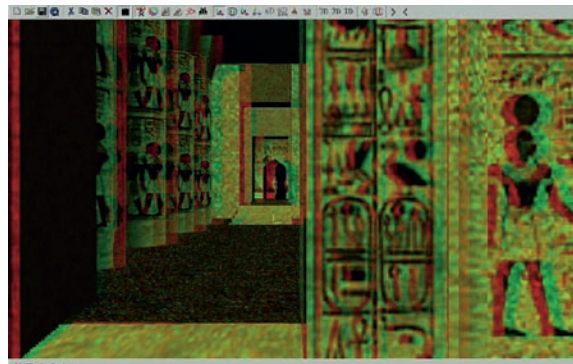
White Dune – продолжение исходного проекта *Dune*, предназначенного для упрощения загрузки, редактирования и анимирования файлов VRML – по сути, 3D-миров, загружаемых в любом совместимом браузере.

За время разлуки возникло множество функций. Добавлена зачаточная поддержка X3D (преемника VRML97), а последняя бета-версия обзавелась возможностью вызова движка анимации из C++.

Лично мне интереснее всего было узнать об интерфейсе 4kids – соответствующий значок вы сразу увидите, установив *White Dune* – который разбивает зачастую сложный интерфейс на несколько более

простых опций; простых настолько, что даже ребенок (или умственно отсталый журналист) справится с VRML-анимацией.

Сам-то я не создал ничего особо примечательного или даже сносного, но если *Blender* или *Lightwave* раньше показались вам слишком сложными, стоит обратить внимание на *White Dune*.



» *White Dune* имеет даже режим красного/зеленого 3D-ореола, так что вы сможете придать большую глубину своим творениям.

Образ и клон диска

Clonezilla

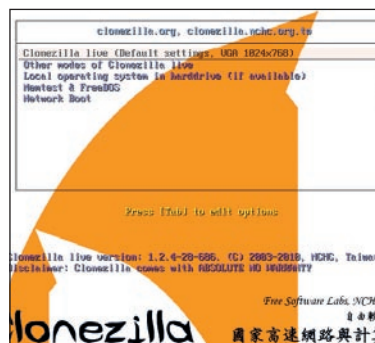
Версия 1.2.8-42 Сайт www.clonezilla.org

Программы Symantec – мощная вещь; *Norton Ghost*, проприетарная и коммерческая, столь хороша для систем клонирования, что слово «ghost» вошло в популярный лексикон для обозначения соответствующего действия.

Но у *Ghost* есть недостатки. А именно: версия для обычной публики одноканальная. Один диск копируется в один файл, а затем вы переносите файл на машины, где хотели создать клон. Копировать сразу на много машин способно только корпоративное издание, но даже и в этом случае его коммерческая природа (не говоря уж о стоимости) больно ранит наши чувства.

Clonezilla, основанная на *DRBL*, *Partclone* и *udpcast*, делает то же самое и не стоит ни копейки. Подобно своему вдохновителю, она имеет пару версий: версию Live, которая эффективно создает копию одной системы, и SE, редакцию для сервера, которая – при условии, что вы настроили соответствующую машину как сервер –

умеет писать образ диска на множество машин сразу. Незачем даже устанавливать специальный сервер; имея копию образа диска, который вы хотите клонировать, запускайте *DRBL* live и работайте оттуда. Разработчик-тайванец умудрился одновременно установить на 40 компьютеров 5-ГБ образ примерно за 30 минут, так что это – достаточно крепкий инструмент для практически любого клонирования. LXF



» Загрузите *Clonezilla* с live CD, и получите это простое меню. Этот инструмент при использовании не упрямится.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

» **Proelia 1.0.2**

Отслеживайте рисованные RPG-игры всех форм и размеров. www.pilgrimagesoftware.com/products/proelia/

» **OpenDIAS 0.6.4**

Сканируйте много документов? Организуйте вывод для распознавания текста этим удобным инструментом. www.essentialcollections.co.uk/openDIAS/

» **Magic Boot USB 2011-05-14**

Попробуйте этот интересный прием, чтобы заставить несговорчивые машины загружаться с USB. www.guciek.net/magicbootusb/en



» Создайте игру в пурпурных и зеленых тонах. Ох, нет. Проявите больше вкуса.

» **Cottage 1.2.0.beta1-1**

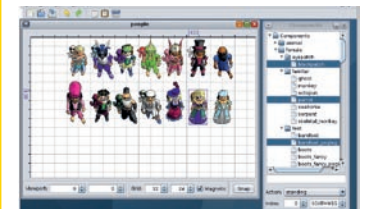
Выкачивайте HTML5-совместимые 3D-среды без проблем. www.anemonesoft.com/os/cottage/

» **Motorcycle Blackjack 0.1**

Заждались игры Blackjack с темой Honda CBR250R? www.freshmeat.net/projects/motorcycle-blackjack

» **Scenepainter 3.4**

Создавайте комиксы для личного пользования, используя широкий выбор готовых элементов. <http://scenepainter.onyxbits.de/>



» Просто выбирайте персонажей в меню слева.

» **GeoToad 3.14.4**

Геокэшинг не обязан быть скучным. Используйте его для создания простых запросов. www.code.google.com/p/geotoad/

» **QSimpleGuitar 0.1**

Забудьте о камертонах и дорогой электронике – вот вам инструмент. www.code.google.com/p/qsimpleguitar/

На диске

Дистрибутивы, игры, приложения... и опять дистрибутивы...



Прочтите всю информацию, необходимую для использования DVD! Если вы новичок в Linux, откройте файл `index.html` на диске и перейдите в раздел Справка, где вы найдете руководства по вопросам, перечисленным справа:
Майк Сондерс, редактор диска
mike.saunders@futurenet.com

- » Что такое Linux?
- » Что такое дистрибутив?
- » Загрузка ПК с DVD
- » Разбиение жесткого диска на разделы
- » Навигация по файловой системе
- » Использование командной строки
- » Установка программ

Дистрибутивы Linux

Arch, Debian, Mint, SUSE, Ubuntu

На специальном двустороннем DVD этого месяца – сущий праздник дистрибутивов. На Стороне 2 – полная версия Fedora 15, загружаемая прямо с диска и включающая тысячи пакетов. Вы прочтаете о нем все и познакомитесь с руководством по установке, перевернув страницу. На Стороне 1 также есть и другие дистрибутивы, о которых рассказывалось в нашем материале. По возможности мы постарались сделать эту массовую загрузочку с одного и того же диска, но из-за технических ограничений некоторые включены в виде ISO-образов.

Первым идет Arch, многогранный дистрибутив, нацеленный на пользователей Linux среднего и продвинутого уровня. Он не поведет вас за ручку по всем стади-

ям установки; напротив, он предполагает, что вы сами в курсе того, что делаете. Arch – возобновляемый дистрибутив, так что вместо обновления раз в несколько месяцев вы просто качаете новые пакеты по мере их появления.

«Debian 6, несомненно, самая добротная разновидность Linux.»

Для упрощения установки команда Arch выдает в доступ моментальные снимки базовой системы; они-то и находятся на Стороне 1 **LXF DVD**. Загрузите диск и выберите их в меню – меню загрузится в текстовом режиме, предоставляя вам информацию при запуске программы установки на базе *Ncurses*.

Второй загружаемый дистрибутив – Debian 6, несомненно, самая добротная разновидность Linux и чемпион нашего дистрибутивного матча. Debian раньше подвергался критике за некоторую старомодность и за-

косность, но Debian 6 ввел в дистрибутив современную технологию Linux. Со Стороны 1 загружается Debian Live с легковесным рабочим столом LXDE, а захотите установить его на свой жесткий диск – там есть значок программы установки.

И, наконец, третий загружаемый дистрибутив – OpenSUSE. Это давний дистрибутив, хорошо известный как демонстрационная площадка новых релизов KDE. OpenSUSE тоже загружается в режиме live со Стороны 1 – можно познакомиться с ним, не устанавливая на свой жесткий диск, но на рабочем столе опять-таки предусмотрен значок программы установки.

Linux Mint и Ubuntu идут как ISO-образы в разделе Distros. Можете записать их на CD-R и загрузиться с них – обратите внимание, что их надо записывать как образы ISO, а не просто копировать ISO на диск. При проблемах гляньте в документацию вашей программы для записи дисков. Три оставшиеся ОС, которым тоже посвящены материалы нашего журнала (*FreeNAS*, *Damn Small Linux* и *Puppy Linux*), также находятся в разделе Distros.

Важно

ВНИМАНИЕ!

» Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

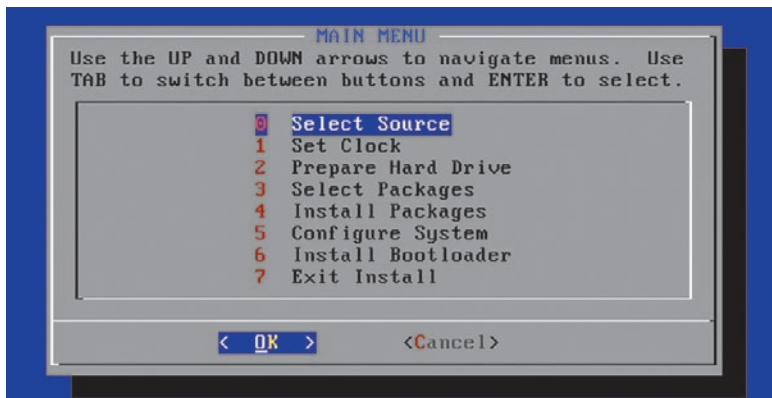
Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

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 лицензию.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.



» Программа установки Arch Linux не отличается графическими изысками других дистрибутивов, но при некоторых базовых знаниях в Linux она не сложна.

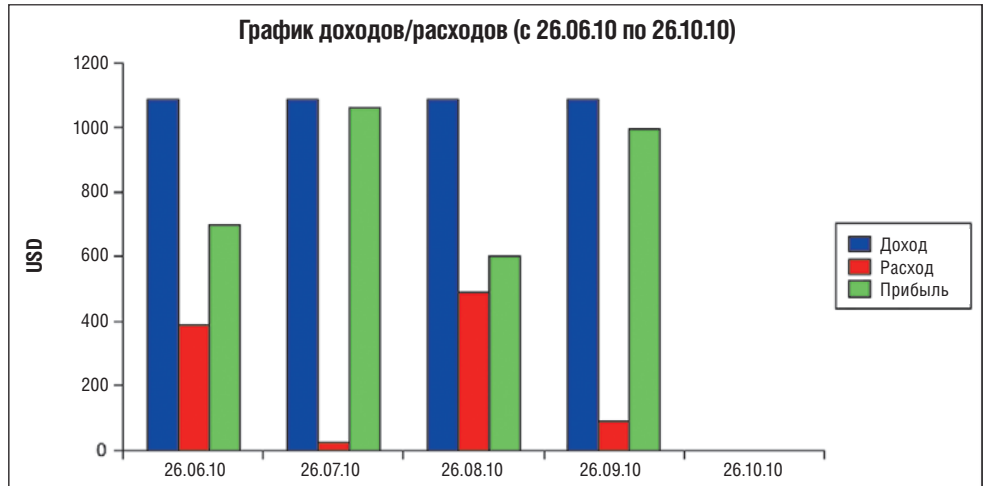
Другие программы

GnuCash, Qtractor, средства разработчика...

Но DVD этого месяца – не только парад дистрибутивов. Открыв [index.html](#) на Стране 1, вы найдете отличный выбор новых программ. *GnuCash* в разделе Desktop – один из лучших пакетов для управления персональными финансами, позволяющий отслеживать банковские счета, акции и расходы и совершать прочие головокружительные трюки с цифрами, чтобы следить за своими деньгами. У него очень деятельные разработчики, и обновления и исправления ошибок выходят раньше любых ожиданий; их компания – настоящая звезда сообществ свободных программ.

А вот *Qtractor* – мультитрековый аудиосеквенсор, весьма эффективно применяющий ALSA и JACK. Он создавался как «рабочая станция аудио» и «персональная домашняя студия»; интерфейс слегка пугает начинающих пользователей (впрочем, как в большинстве аудиоприложений), зато функций в нем полно. Он поддерживает секвенсирование и запись MIDI, неограниченное число треков в проекте, взаимодействие через drag-and-drop и поддержку модулей расширения LADSPA, DSSI, VSTi и LV2. И даже сейчас, в версии 0.4.9, это очень полезное приложение.

В разделе Development имеется *Qt Creator*, самая свежая IDE *Qt* от Nokia – вместе с XBPS, новой попыткой замахнуться на двоичную систему пакетов. Затем здесь есть подборка языков: Chicken переводит код Scheme в «относительно портативный» C, а Interp может работать на встроенных платформах, на которых нужен только серийный порт – и больше ничего.



Amforth – еще один встраиваемый язык, реализующий FORTH для чипов Atmel AVR ATmega – как ни поразительно, для работы ему требуется всего 8К ОЗУ!

В заключение – о наших постоянных рубриках. В разделе Help вы найдете Answers Archive (Архив Ответов), подборку более 700 проблем Linux и их решений, из предыдущих выпусков журнала. Если у вас проблемы с установленным на компьютере Linux, откройте этот раздел в своем браузере и примените по назначению его функцию поиска по ключевым словам.

В разделе Magazine на Стране 1 помещен исходный код для руководств в нашем приложении по программированию, а также самые свежие выпуски подкаста Tux-Radar в формате OGG и MP3. Если раньше вы его не слушали – этот подкаст подготовлен теми самыми людьми, что создают

для вас этот журнал: каждые две недели мы рассаживаемся вокруг микрофона, чтобы обсудить самые горячие новости Linux, ну и, конечно, болтаем не по существу. Послушайте!

► Графики *GnuCash* позволяют вам с первого взгляда оценить состояние своих финансов.

И кое-что еще!

Четыре крутых игры...

Некоторые великолепные игры не умеют добиваться заслуженного ими признания, и *Pushover* – типичный тому пример. Эта головоломка на основе домино была широко известна в 90-е, и концепция осталась той же: вам нужно заставить упасть все костяшки домино на уровне, запустив цепную реакцию. Некоторые из них взрываются, некоторые бросают вызов гравитации, а некоторые делятся надвое. Уровни выглядят потрясающе, и играть очень весело, так что мы очень рады этому ремейку (с тем же названием). Вы найдете исходный код, компилируемый

обычным порядком – `./configure`, `make` и `make install` (последняя строчка – от имени root), в разделе Games.

Следом идет *Hydra Slayers*, *Rogue*-подобная игра, «вдохновленная математическими головоломками о храбрых героях, убивающих многоголовых чудовищ». Любители подобных забав ее оценят. А *FreeLords* – пошаговая стратегия, похожая на *Warlords*, тогда как *Tetzie* – остроумный вариант головоломки: вы даете ему изображение, он разбивает его на *Tetris*-образные кусочки, а вы собираете изображение заново.



► Загрузочное меню Страны 1 придерживается графики Debian, но предоставляет также и Arch, и OpenSUSE!



► Подурчитесь с домино в *Pushover*.

На диске

Fedora получает совершенно новый рабочий стол с Gnome Shell

Дистрибутив Linux

Fedora 15

Fedora часто лидирует по части реализации новых технологий, и хотя это не первый дистрибутив, принявший Gnome 3, он опередил основных игроков. Для новичков в мире Linux: Fedora – мощный, всесторонний дистрибутив Linux для настольных ПК, рабочих станций и серверов, с тысячами программ. Он задействует передовую технологию Linux и поддержан дружелюбным и работающим онлайн-сообществом, с Red Hat во главе.

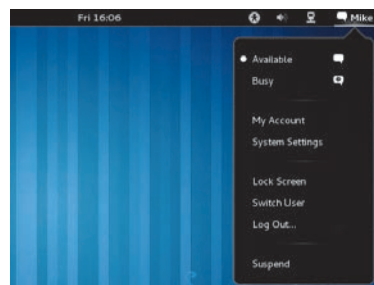
Самые примечательные новые функции в этом релизе – уже упомянутый рабочий стол Gnome 3 (с Gnome Shell); systemd, обеспечивающая последовательность загрузки; динамический брандмауэр (позво-

ляющий вводить изменения без перезапуска); улучшенное управление питанием через *PowerTOP*; и *LibreOffice*. Это смелый релиз со множеством новых функций – но уже поступившие отзывы из Интернет сообщают также и о его надежности.

В специальный двусторонний DVD этого месяца вошла полномасштабная версия Fedora весом 3,5 Гб. Она включает почти 3000 пакетов, предоставляя практически все необходимое – будь вы домашним пользователем настольного ПК, разработчиком или администратором сервера. Если ваш дистрибутив уже успел набить вам оскомину, это отличная возможность воспользоваться всеми благами Linux.

Однако передовая технология имеет свою цену, и эта цена – системные требования: Fedora 15 нужно не менее 768 Мб ОЗУ, и мы рекомендуем минимум 1,5-ГГц CPU и 10 Гб на жестком диске для нормальной работы. Вы можете загрузить программу установки со Стороны 2 *LXF DVD* и выбрать загрузочные опции из меню (там есть пункт загрузки в безопасном видеорежиме на случай, если ваша видеокарта не распознается).

Многие пользователи захотят установить Fedora вместе с другой ОС, например, Windows. (Не переживайте, мы вас не виним: мы же понимаем, что человеку иногда нужна сессия *World of Warcraft*.) В таком случае программа установки Fedora может



➤ Зайдя в настройки системы, переведите машину в режим ожидания (или выключите, нажав Alt) через меню сверху справа.

сжать имеющийся раздел Windows, выделив место. Обычно это проходит без проблем, но, как и при любой серьезной операции с жестким диском, настоятельно советуем предварительно сделать резервные копии важных данных Windows.

Желаете настроить деление на разделы вручную – пожалуйста. Как именно вы решите нарезать свой диск, зависит от установленных вами ОС, но рекомендуем отвести не менее 10 Гб под раздел *root (/)* в формате *ext4*, и создать раздел *swap* размером вдвое больше объема ОЗУ (но не более 2 Гб). Вы всегда можете поместить директорию */home* на отдельный раздел и поделить ее между дистрибутивами, но помните, что разные версии программ могут привести к конфликту разновозрастных версий файлов настройки.



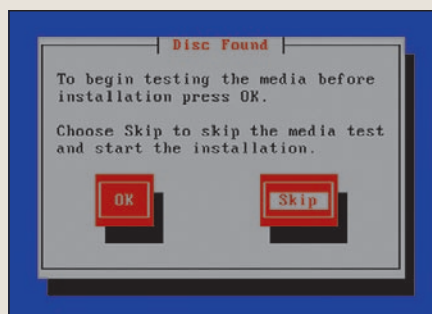
➤ Навигация по открытым окнам и значкам быстрого запуска программ осуществляется щелчком на Activities сверху слева.

Шаг за шагом: Устанавливаем Fedora 15



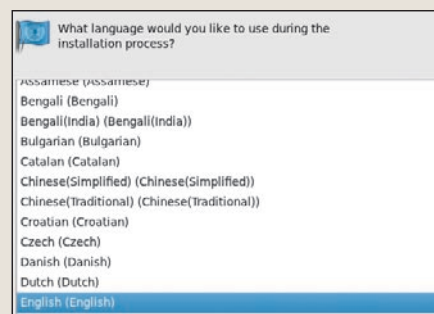
1 Загрузка

Загрузите свой ПК со Стороны 2 *LXF DVD* и нажмите на Enter в меню. (Если при загрузке возникли проблемы, откройте [index.html](#) на Стороне 1 диска и загляните в Help/New to Linux.)



2 Проверка

Стадию проверки диска можно опустить, но если у вас опять появятся проблемы, рекомендуем перезагрузить машину и все-таки провести проверку.



3 Программа установки

Загрузится программа установки и задаст вам вопросы о выборе языка, раскладки клавиатуры и сети.

Как бы мне?..

Если вы только начинаете знакомство с Linux, после установки Fedora у вас могут возникнуть вопросы насчет выполнения определенных задач. Вот список наиболее общих задач и способов их решения...

» **Работа в сети** Щелкните по Activities и оранжево-голубому значку с земным шаром на левой панели, чтобы запустить Firefox, второй по популярности в мире браузер, поддерживаемый сотнями расширений.

» **Работа с файлами** Щелкните по значку шкафчика для файлов в доке Activities, чтобы получить доступ к своим персональным документам.

» **Воспроизведение музыки и видео** В Activities > Applications > Sound & Video вы найдете выбор видео- и аудиоплееров.

» **Редактирование изображений** Перейдите в Activities > Applications > Graphics > GNU Image Manipulation Program, чтобы запустить очень многогранное приложение, имеющее много общего с Photoshop от Adobe.

» **Чат онлайн** В категории Internet вы найдете Empathy, блестящий онлайн-пейджер, поддерживающий все популярные протоколы – MSN, Jabber и Google Chat.

» **Настройка системы** Щелкните по имени пользователя вверху справа, а затем по System Settings – там вы найдете выбор параметров и утилит настройки.

» **Поиск новых программ** В Activities, перейдите в Applications > System Tools > Add/Remove Software, чтобы скачать новые программы из Интернета.

» **Выключение** Щелкните по имени пользователя вверху справа, чтобы вызвать меню, и нажмите на Alt, чтобы превратить Suspend в Power Off.

Если вам понадобится помощь по Fedora, первое место, куда стоит заглянуть – отличный сайт дистрибутива, www.fedoraproject.org. Там в верхней части страницы имеются ссылки для взаимо-



действия с сообществом и получения помощи. Есть даже список основных вопросов, задаваемых новыми пользователями Fedora, и вы также можете осуществлять поиск по документации.

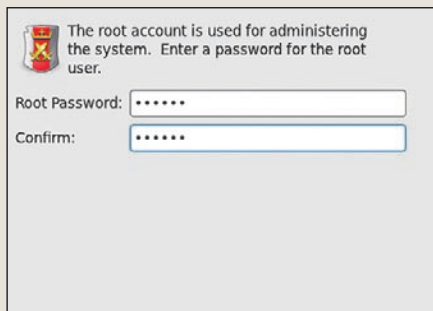
Если у вас возникнет специфический вопрос по дистрибутиву, загляните на www.fedoraforum.org – это отличный ресурс, где более 4000 активных членов, готовых помочь. Четко сформулируйте свою проблему, указав, на каком оборудовании работаете, и кто-нибудь да ответит. Также наведите к нашей дружной команде на www.linuxformat.com/forums/ – хотя бы просто поболтать! LXF

» **Gnome 3 гораздо более минималистский: почти все доступно из Activities.**

Не пропустите...

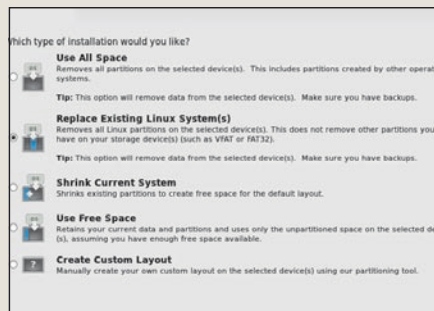
Док
Щелкните правой кнопкой по значкам в доке Favorites, чтобы работать с ними, или удалите их.

LibreOffice
Как и большинство основных дистрибутивов, Fedora перешла на LibreOffice, отвлечение OpenOffice.org.



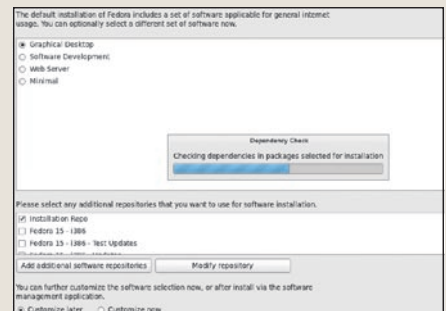
4 Пароль

Вам предложат указать пароль для учетной записи root (admin). Он жизненно важен, так что не забудьте его; да помните, что в пароле важен регистр.



5 Разбиение на разделы

Fedora спросит, как поделить ваш жесткий диск. Можно отвести весь диск целиком под Linux, разделить его с Windows или разбить на разделы вручную (но это – для экспертов).



6 Выбор

Выберите состав программ, которые вы хотите установить (отлично подойдет выбор по умолчанию), затем начнется копирование файлов, после чего вы загрузитесь в свою новую систему.

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>LXF144 Май 2011</p> <ul style="list-style-type: none">» MythTV Лучшая цифровая ТВ-платформа в подробностях» Графопостроители Как осмыслить серые колонки цифр» Zeitgeist Невидимый секретарь, который записывает каждый ваш шаг» Управление проектами Организовать вольных разработчиков ПО трудно, но не невозможно <p>LXFDVD: OpenSUSE 11.4, Puppy 5.2, Mythbuntu 10.10, Wordpress 3.1, игры и прочее</p> <p>Печатная версия: shop.linuxformat.ru/lxf_144/ PDF-версия: shop.linuxformat.ru/elxf_144/</p>	 <p>LXF145 Июнь 2011</p> <ul style="list-style-type: none">» Gnome и GTK затевают революционные преобразования» Музыкальные проигрыватели У меломанов глаза разбегаются» Скажи-ка, дядя... Так говорит Ричард Столлмен» Calligra Офисных комплектов много, но этот рвется в мобильники» Компьютер в розетке Размером не больше электробритвы <p>LXFDVD: Jolicloud, Arch Linux, Gnome 3, Trisquel, игры и прочее</p> <p>Печатная версия: shop.linuxformat.ru/lxf_145/ PDF-версия: shop.linuxformat.ru/elxf_145/</p>	 <p>LXF146 Июль 2011</p> <ul style="list-style-type: none">» Ubuntu 11.04 в корне изменил стиль работы» Оконные менеджеры-легковесы Зачем вам лишние роскоши?» KDE становится проще с Plasma Active» Жить в сети Google ввел в нетбук кнопки для web-серфинга <p>LXFDVD: Ubuntu 11.04 и компания, Firefox, Tiny Core 3.6, игры и прочее</p> <p>Печатная версия: shop.linuxformat.ru/lxf_146/ PDF-версия: shop.linuxformat.ru/elxf_146/</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 озадачены этим вопросом, 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.

Август 2011
LXF DVD 147LINUX
FORMATАвгуст 2011
LXF DVD 147LINUX
FORMAT

fedora 15



» Ядро 2.6.38.2 » Gnome 3 с Gnome Shell » KDE 4.6 » Xfce 4.8

Более 2900 пакетов
32-разрядная сборка

А ТАКЖЕ: LibreOffice 3.3.2, Firefox 4.0, Asterisk 1.8 и свежие версии остальных пакетов

Подборка
дистрибутивов

» Смотрите их сравнение в «Матче дистрибутивов» на с. 20



» Загрузка с DVD: Arch Core, Debian 6, OpenSUSE 11.4
» ISO-образы: Damn Small 4.4, Linux Mint 10, Puppy Linux, Ubuntu 11.04

А ТАКЖЕ: FreeBSD 8.0, SuseLinux 11.2.8, средства разработки 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 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 Linux** Базовая система, для последующей доводки из репозитория (загрузка с **LXF DVD**)
- Debian Small 4.4.10** Дистрибутив с минимальными требованиями к аппаратуре (ISO-образ)
- Debian 6** Актуальная версия популярного дистрибутива (загрузка с **LXF DVD**)
- FreeNAS** Специальный дистрибутив для построения сетевых устройств хранения данных (ISO-образ)
- Linux Mini 10** Дистрибутив на базе Ubuntu, но с «классическим» боюте (ISO-образ)
- OpenUSE 11.4** Актуальная версия популярного дистрибутива (загрузка с **LXF DVD**)
- Puppy Linux 5.2.5** Популярный «легкий» дистрибутив (ISO-образ)
- Ubuntu 11.04** Актуальная версия популярного дистрибутива (ISO-образ)
- Fedora 15 DVD** Полная версия, 32-разрядная сборка (вторая сторона **LXF DVD**)

РАБОЧИЙ СТОЛ

- din 1.6.1** Музыкальный синтезатор
- GnuCash 2.4.5** Программа управления персональными финансами
- Kdenlive 0.8** Программа для редактирования видео
- Lim 2.3** Файловый менеджер, написанный на Python и cgiPerl
- 0002GD 2.4.0** Средство импорта/экспорта документов
- Qtractor 0.4.9** Приложение для сведения аудио/МIDI-треков
- RPD 0.4.1** Программа для импорта фотографий с камер и с других носителей
- Task Coach 1.2.18** Планировщик и менеджер заданий
- vxCam** Программа для работы с веб-камерами; поддерживает драйверы видеофайлов 11.2

РАЗРАБОТКА

- amforth 4.4** Реализация Forth для микроконтроллеров Alpha AVR ATmega
- Chicken 4.7.0** Компилятор схем RRSRS в код C++
- interp 01.08.0** Язык и система программирования для встроженных систем
- Qt Creator 2.2.0 DE** Для создания приложений Qt
- XFPS 0.8.0** Набор библиотек для работы с X

ИГРЫ

- FreeLords 0.4.2** Свободная реализация игры Warlords
- Hydra Slayer 11.3** Ворче-подобная RPG, борьба с многоголовыми гирями
- Pushover 0.0.3** Головоломка в стиле Sokoban
- Tetris 2.0.0** Головоломка-пазл

НОТПЭКС

- Abuse 0.8** Игра, пошаговая стратегия
- Clonezilla 1.2.8** Средство для клонирования дисковых разделов по сети, Live CD
- Conquests 0.12** Игра, пошаговая стратегия
- LIRC 0.9.0** Программа для дистанционного управления устройствами по ИК-каналу
- Lux 2.0** Редактор документов TeX/LaTeX
- mlWaveEdit 1.4.21** Программа для записи, редактирования и проигрывания аудиофайлов
- Rekonq 0.7.0** Браузер для KDE, использующий WebKit
- SwarmTV 0.9.2** Демон, автоматически отслеживающий RSS и загружающий torrent-файлы
- WeeChat 0.3.5** Кросс-платформенный клиент чата
- White Dune 0.3 beta** Ныксовременный редактор X3D/VRML7

INTERNET

- aria2 1.11.2** Менеджер загрузок, поддерживающий HTTPS), FTP, BitTorrent и Metalink
- BoGoMa 1.0** Средство резервного копирования для Google Mail
- Barfroot 1.0** Средства балансировки нагрузки для входящего TCP-трафика
- JunKieTheSniffer 1.1** Анализатор пакетов, работающий в реальном времени
- ssinfo 0.0.41** Программа для мониторинга состояния компьютеров в локальной сети

СИСТЕМА

- CUPS 1.5b2** Стандартная система печати для Linux (исходные коды)
- Dnswaver 1.2.0** Универсальный SQL-клиент и менеджер СУБД, работает через JDBC
- G4L 0.3.7** Программа для клонирования дисковых разделов
- phpVirtualBox 4.0-6** Пользовательский интерфейс VirtualBox, реализованный на Ajax и PHP
- strongSwan 4.5.2** Реализация IPsec для Linux, Android, Maemo, FreeBSD и Mac OS X

Пождауйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕЯТЕЛЬНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех старых платформах, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать адекватный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных на нем программ или данных. Прежде чем устанавливать какие-либо ПО на компьютер, пожалуйста, посетите www.linuxformat.ru за сведениями об авторских правах.

Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАР 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.

IT-архитектура вашего бизнеса



119991 Москва, ул. Губкина, д. 8
Телефон: +7 (495) 232-00-23
Электронная почта: info@softline.ru
Сайт: www.softline.ru

softline[®]

19 стран, 61 город



Школа LXF

Спонсор рубрики
Mandriva.ru
разработчик
дистрибутива
EduMandriva
www.mandriva.ru

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Контроль издалека

Татьяна Казанцева рассказывает, как пресечь «левую» деятельность школьников на уроке, не отходя от кас... учительской кафедры.



Наш
эксперт

Татьяна
Казанцева

в свободное время от корпения над написанием методики скрещивания Scratch и Arduino оттачивает навыки работы со свободным ПО для использования в школе и дома.

«Высоко сижу, далеко гляжу», говорила Маша из небезвестной сказки, пытаясь показать медведю, что она контролирует процесс доставки пирожков. А можно ли проделывать такое в классе и проследить, что происходит на машинах учеников, сидя за учительским компьютером?

Как говорится на IT-сленге, «хороший админ – ленивый админ». Давным-давно умные деятели компьютерных наук поняли, что бегать к пользователям по каждому зову – занятие неблагодарное, и придумали способы, позволяющие управлять машинами удаленно. А нельзя ли повернуть такое в компьютерном классе и сделать даже больше – увидеть, чем заняты ученики во время урока: не играют ли в какие-нибудь игры вместо написания задачи на нелюбимом языке программирования и не пытаются ли разукрасить рабочий стол в ультрамодный розовый цвет?

Ответ, как всегда, будет положительный (иначе не было бы этой Школы LXF). Вы не только можете увидеть паранормальную активность на рабочих местах, но и непосредственно поучаствовать в этом процессе, перехватить управление и сделать еще больше. И все это – как в текстовом (вдруг среди вас есть ретрограды, которые боятся всего, что отличается от командной строки), так и в графическом режиме.

Пользователи «других форточек» могут сказать, что у них уже давно есть RAdmin и прочие примочки для несанкционированного доступа на удаленный рабочий стол; но в том и прелесть свободного ПО, что в Linux этих возможностей в разы больше, и вы даже сможете залезть к «этим форточникам» на их компьютеры и поуправлять ими также.

Давайте по порядку рассмотрим, что же может противопоставить Linux пылливому и одновременно изворотливому уму школьника.

О чем умалчивает ssh

При первом заходе админа в сеть он сразу же попытается сделать две вещи: заблокировать пользователям способы лезть куда не надо и настроить себе доступ по SSH ко всем имеющимся машинам. Про SSH *LinuxFormat* опять же писал неоднократно (см. врезку), и это весьма удобная вещь, позволяющая получить доступ к удаленной командной строке с достаточной безопасностью.

Настроить такой доступ довольно просто. Первым делом вам нужно на машинах учеников (да-да, именно на них) установить необходимые пакеты сервера SSH. Я буду все рассматривать на примере любимого (и одного из лучших, по моему мнению) дистрибутива EduMandriva (многие, конечно, будут шуметь, почему не Ubuntu, но тут дело моего вкуса), для которого нужно установить следующий пакет – *openssh-server*.

На машине учителя, следовательно, должен быть клиент доступа – *openssh-client*.

После установки не забудьте запустить демон *sshd*, дав команду

```
/etc/init.d/sshd start
```

Далее вам нужно настроить доступ по ключам. Можно, конечно, разрешить доступ на машины пользователю *root*, отредактировав строку в */etc/ssh/sshd_config* и установив

```
PermitRootLogin yes
```

но это небезопасно. Лучше будет выполнить описанные далее шаги. Первым делом создайте пару файлов ключей с помощью команды

```
ssh-keygen
```

У вас запросят парольную фразу. Оставьте ее пустой. Теперь в каталоге *~/.ssh* есть два файла: *id_rsa* и *id_rsa.pub*. Первый файл – это закрытый ключ, и его нужно хранить в безопасном месте, а второй – открытый, его нужно добавить в файл *~/.ssh/authorized_keys* на удаленном компьютере. Файл можно скопировать на флэшку, вставить ее в удаленный компьютер и добавить ключ с помощью команды

```
cat id_rsa.pub >> ~/.ssh/authorized_keys
```

Теперь разрешите аутентификацию по ключу, установив параметр

```
PubkeyAuthentication yes
```

в файле */etc/ssh/sshd_config* на удаленном компьютере, и перезапустите сервис SSH:

```
/etc/init.d/sshd restart
```

Отключите вход по паролю добавкой в файл *sshd_config* следующих строк:

```
PasswordAuthentication no
```

Если вы все сделали правильно, то команда

```
ssh user@192.168.1.10
```

позволит вам зайти на машину с IP-адресом 192.168.1.10 от имени пользователя *user*.

Чтобы иметь возможность запуска графических приложений с этой машины, сделайте X-проброс – для этого нужно, чтобы в файле *sshd_config* была установлена опция

```
X11Forwarding yes
```

– и запускайте *ssh* с ключом **-X**.

```
ssh -X user@192.168.1.10
```

Если вас не пускают, не забудьте проверить настройки брандмауэра. У вас должен быть открыт порт 22 (или разрешен доступ по SSH при настройке из графического режима).

Ну и в самом плачевном случае – если вы не понимаете, что тут написано, дайте эту статью способному ученику (или его папе), и пускай он поможет вам с настройкой.

Теперь – само собой разумеющийся вопрос: чего ради эти мучения? Как это поможет контролировать работу учеников? А очень просто – мы получили удаленный доступ к командной строке пользователя, то бишь ученика, и можем делать оттуда все, что захотим.

Давайте попробуем проконтролировать, тем ли он занят на своей машине. Для этого, выполнив вход на удаленную машину, введите следующую команду (мы будем подразумевать, что имя пользователя на всех учебных машинах – user):

```
ps -u user
```

Это выведет нам список всех процессов, запущенных от имени данного пользователя. Вы должны получить что-то типа такого:

PID	TTY	TIME	CMD
5050	?	00:00:01	lxsession
5115	?	00:00:00	ssh-agent
5140	?	00:00:01	gpg-agent
5251	?	00:00:00	dbus-launch
5252	?	00:00:00	dbus-daemon
5265	?	00:00:00	s2u
5271	?	00:00:27	openbox
5275	?	00:01:50	lxpanel
5277	?	00:00:19	pcmanfm
5278	?	00:00:10	xscreensaver
5280	?	00:01:33	parcellite
5282	?	00:00:05	nm-applet
5285	?	00:00:01	menu-cached
5288	?	00:00:00	gnome-keyring-d
5295	?	00:00:00	gvfsd
5302	?	00:00:02	gconfd-2
5304	?	00:00:00	volumeicon
5307	?	00:00:00	gnome-keyring-d
5315	?	00:03:56	pulseaudio
5329	?	00:00:00	gvfs-fuse-daemo
5338	?	00:00:00	gconf-helper
5552	?	00:00:02	gvfs-gdu-volume
5560	?	00:00:00	gvfs-gphoto2-vo
7704	?	00:06:15	chrome
8989	?	00:02:40	soffice.bin
9985	?	00:03:50	lxterminal
9986	?	00:00:00	gnome-pty-helpe
15200	pts/1	00:00:00	bash
15985	?	00:00:00	gvfsd-trash
18701	?	00:00:01	sol
19568	pts/1	00:00:00	ps

Казалось бы, что можно узнать из перечня непонятных названий? Для достижения понимания введите эту же команду для своего (учителя) пользователя на вашей машине, запустив предварительно только те приложения, которые необходимы в текущем задании. А затем сравните выводы, отметив различающиеся программы. К примеру, в нашем случае учащиеся должны выполнять задания в *LibreOffice*. Из вывода видно, что на компьютере ученика запущен *soffice.bin*, а значит, он работает в чем полагается (для *OpenOffice.org* процесс будет такой же). Но два процесса будут отличаться – это *chrome* и *sol*. Первый, как несложно догадаться, является браузером Chromium, а вот второй процесс – ничем иным, как пасьянсом *[solitaire]* (скажем, пресловутой Косынкой). Оба эти процесса подлежат уничтожению как не относящиеся к делу.

Названия процессов легко изучить (и записать), запуская программы на своей машине. Последние запущенные процессы будут иметь более высокий PID, и их легко будет отличить от ранее используемых.

Теперь же, распознав, что учащийся занимается «левыми» делами, применим магию командной строки (вы помните, что мы все еще залогинены через SSH) и уничтожим эти вредоносные для

SSH — безопасная командная строка

SSH [англ. Secure Shell – «безопасная оболочка»] – сетевой протокол свансового уровня, позволяющий производить удаленное управление операционной системой. Он шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

SSH позволяет безопасно передавать в незащищенной среде практически любой другой сетевой протокол. Таким образом, можно не только удаленно работать на компьютере через командную оболочку, но и передавать по зашифрованному каналу звуковой поток или видео (например, с веб-камеры). Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например,

для удаленного запуска графических приложений. То есть вы можете сделать так называемый X-проброс, запустив, к примеру, *LibreOffice* на школьной допотопной машине со 128 МБ ОЗУ, хотя реально он может выполняться на удаленном 64-ГБ сервере в соседнем здании.

Подробнее об SSH мы уже писали на страницах **LXF**. Вы можете почитать следующие статьи (их можно найти или в PDF-выпусках журнала, или на вики *LinuxFormat* – wiki.linuxformat.ru):

» *SSH и VNC: Работа издали (LXF119)*

wiki.linuxformat.ru/index.php/LXF119:ssh

» *Часто задаваемые вопросы. Соединяемся (LXF102)*

wiki.linuxformat.ru/index.php/LXF102:Ответы

» *Краткая справка. X-проброс (LXF106)*

wiki.linuxformat.ru/index.php/LXF106:Ответы

данного урока приложения. Для этого достаточно скопировать **killall <имя процесса>** –

```
killall sol
```

– и враг повержен!

Таким же способом можно выключить удаленно и другие программы.

В порядке побочного эффекта вы получите возможность удаленно выключить или перезагрузить (**poweroff** или **reboot**) машину учащегося или, запустив, к примеру, **mc**, посмотреть текст написанной им программы или запустить скомпилированную учебную задачу.

А графически — нельзя?

Но все-таки командная строка для многих преподавателей сродни колдовству, и только посмотрев на экран, они могут понять, что реально происходит на машине. А нельзя ли увидеть экран компьютера ученика? Да, тоже можно. Можно использовать технологии удаленного рабочего стола (к примеру, *VNC*), но лучше применить специализированные программы, которые также позволяют транслировать свой рабочий стол на экраны учащихся; перехватывать управление; блокировать все экраны, привлекая внимание; и т.п. Одной из самых известных программ такого рода является *ITALC* (<http://italc.sourceforge.net/>). Она входит во все известные школьные дистрибутивы, и если вы используете Школьный Линукс, ПС-ПО или Edumandriva, то ее установка будет очень простой – максимум, вам нужно будет внести пользователя в группу **italc** командой **usermod -G italc user**

»

```

fzizk@localhost:~/home/fzizk
[root@localhost fzizk]# ica -createkeypair

creating new key-pair ...
Пн ноя6. 15 14:06:59 2010: [warning] QDir::mkpath: Empty or null file
name(s)
Пн ноя6. 15 14:06:59 2010: [warning] QDir::mkpath: Empty or null file
name(s)
... done, saved key-pair in
/etc/italc/keys/private/teacher/key
and
/etc/italc/keys/public/teacher/key

For now the file is only readable by root and members of group root (
if you
didn't run this command as non-root),
I suggest changing the ownership of the private key so that the file
is
readable by all members of a special group to which all users belong
who are
allowed to use iTALC.

[root@localhost fzizk]#

```

» **Результат работы**
ica -createkeypair.

и программу можно будет запускать. Все остальные компоненты, а именно клиенты *italc-client* (ICA) и управляющий *italc-master*, уже будут установлены и настроены. Но давайте рассмотрим установку и настройку, как если бы мы использовали *iTALC* впервые.

iTALC был разработан именно для использования в школе и предлагает много возможностей для учителей, таких как

- » просмотр, что происходит в компьютерной лаборатории, в режиме Обзор, с возможностью сделать снимки;
- » дистанционное управление компьютерами, для поддержки и помощи другим людям;
- » показ демо (в полноэкранном режиме либо в окне) – вывод экрана учителя на компьютеры всех учеников в режиме реального времени;

» блокировка рабочих станций для перемещения пристального внимания на учителя;

- » отправка текстовых сообщений для учеников;
- » удаленное включение, выключение и перезагрузка компьютера ученика;

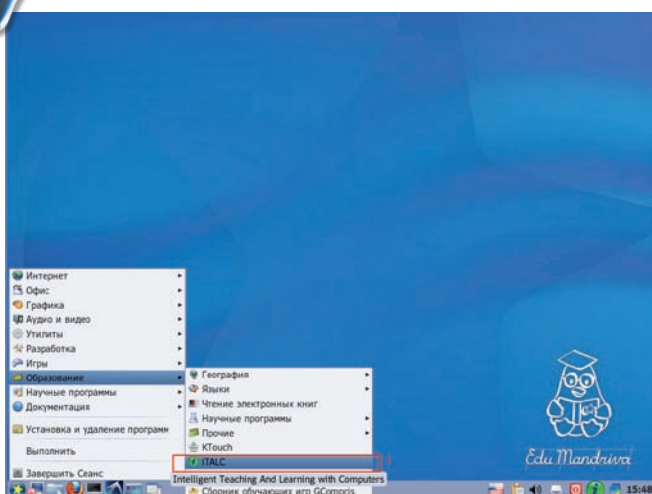
» удаленный вход в систему и выход из системы и удаленное выполнение произвольных команд/скриптов;

» домашнее обучение – сетевая технология *iTALC* не ограничена подсетью, так что ученик может присоединиться к уроку и дома, через VPN-подключение, всего лишь установив клиент *iTALC*.

Итак, мы имеем в наличии компьютерный класс, где *n* ученических компьютеров и 1 компьютер преподавателя объединены в единую локальную сеть. ПО *iTALC* состоит из 2-х частей: клиента (устанавливается на ученическую машину и на компьютер учи-

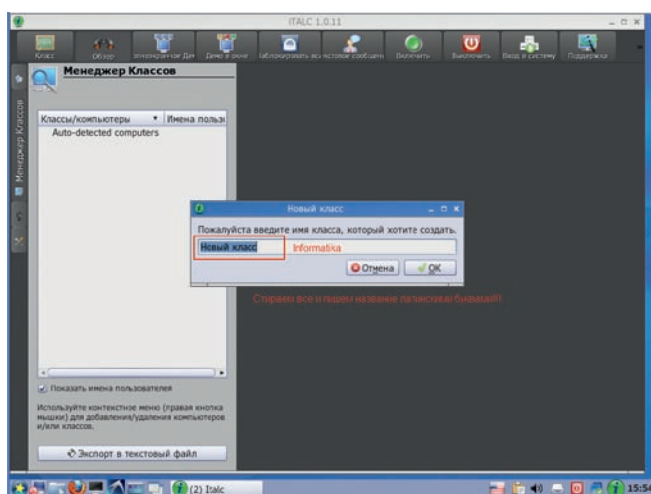


Шаг за шагом: Настраиваем iTALC



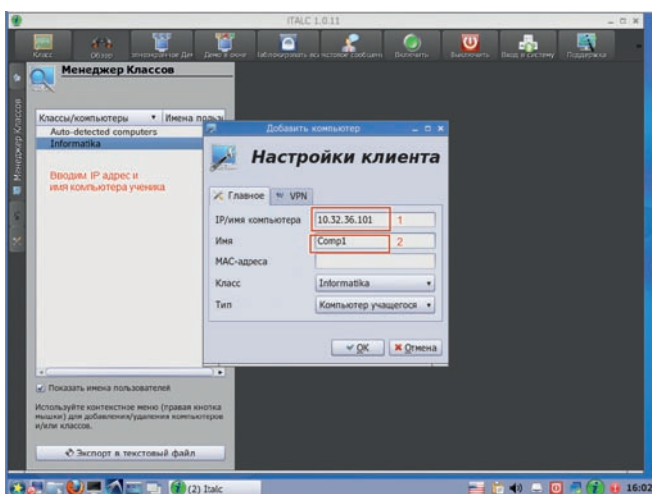
1 Главное — начать

На учительском компьютере запускаем административную часть программы.



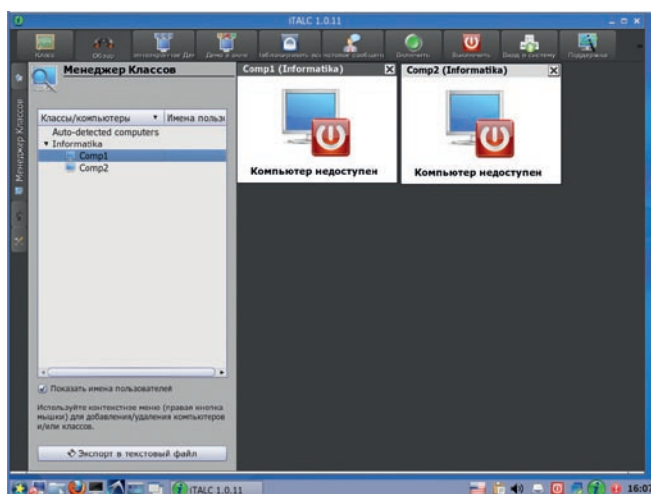
2 Создаем новый класс

Хитрость в том, что вы можете управлять более чем одним классом, и даже под управлением Windows!



3 Добавляем компьютер

учащегося. Перед этим желательно составить схему сети класса с указанием IP-адресов машин.



4 Повторяем добавление

компьютеров для каждого ученического компьютера. Не забудьте экспортировать список в текстовый файл, чтобы не проделывать эту работу каждый раз.

Дважды щелкая по имени компьютера, получаем мини-копию экрана удаленного компьютера (они появляются в левом верхнем углу рабочего поля программы, поэтому их нужно сдвигать на свободное место). Дальше остается изучить основные функции программы. Это можно сделать самостоятельно.

теля) и мастера (устанавливается только на компьютер учителя). Клиент позволяет подключаться мастеру и управлять работой удаленной машины. Мастер содержит интерфейс для управления удаленными компьютерами (компьютерами учеников).

Первым делом добавим пользователя, от имени которого будем работать, в группу **italc**. Потом откроем в брандмауэре порты TCP и UDP 5800–5900 для работы (если у вас разрешены все подключения, этот шаг можно пропустить).

(В Edumandriva эти настройки уже сделаны по умолчанию, поэтому данный шаг пропускаем.)

Создадим ключи: публичный [public] – для компьютеров учеников и приватный [private] – для компьютера учителя. Для этого войдем в консоль администратора (в любом эмуляторе терминала переключимся на root) и выполним команду:

```
ica -createkeypair
```

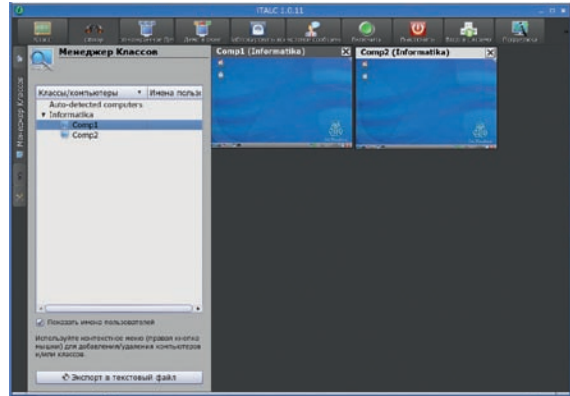
которая сгенерирует необходимые ключи и расположит их в нужных папках.

На рисунке виден отчет генерации и пути к ключам:

/etc/italc/keys/private/teacher/key – приватный ключ (недоступен для пользователя)

/etc/italc/keys/public/teacher/key – публичный ключ (доступен для файловых операций)

Для работы клиента, как и в случае с SSH, нужно скопировать публичный ключ с учительского компьютера на все компьютеры учеников в соответствующую папку. Напомним, он лежит в папке **/etc/italc/keys/public/teacher/key** и не имеет расширения. Этот файл необходимо скопировать на все ученические компьютеры класса в эту же директорию (**/etc/italc/keys/public/teacher/key**). Если в ней имеется ключ, то его необходимо перезаписать но-



► В результате вы должны получить картинку типа вот этой.

вым. Чтобы скопировать ключ на ученический компьютер, нужно войти как root, в противном случае система не даст перезаписать файл ключа. Эти действия нужно повторить на каждом компьютере ученика. Затем настраиваем **iTALC**.

После этого вы спокойно можете контролировать работу учащихся, выдавать на их компьютеры изображения со своего («полноэкранный демо»), делать картинку-в-картинку, чтобы учащийся мог повторять за вами действия («демо в окне»). Щелкнув два раза на окне учащегося, вы сможете подробнее рассмотреть, что он делает, и в случае необходимости даже перехватить управление. Попробуйте, это очень просто!

Напишите нам, если вы чего-то не поняли или хотите, чтобы мы рассмотрели какие-то моменты поподробнее. Мы ждем ваших откликов. **LXF**

Академическая программа для учебных заведений

Mandriva Linux

Mandriva.Ru предоставляет учебным заведениям лицензию, дающую право на неограниченное по числу рабочих станций использование дистрибутива Mandriva Linux на всех компьютерах в образовательном учреждении, всех компьютерах преподавателей и всех компьютерах учащихся, в том числе и домашних.

Комплект поставки:

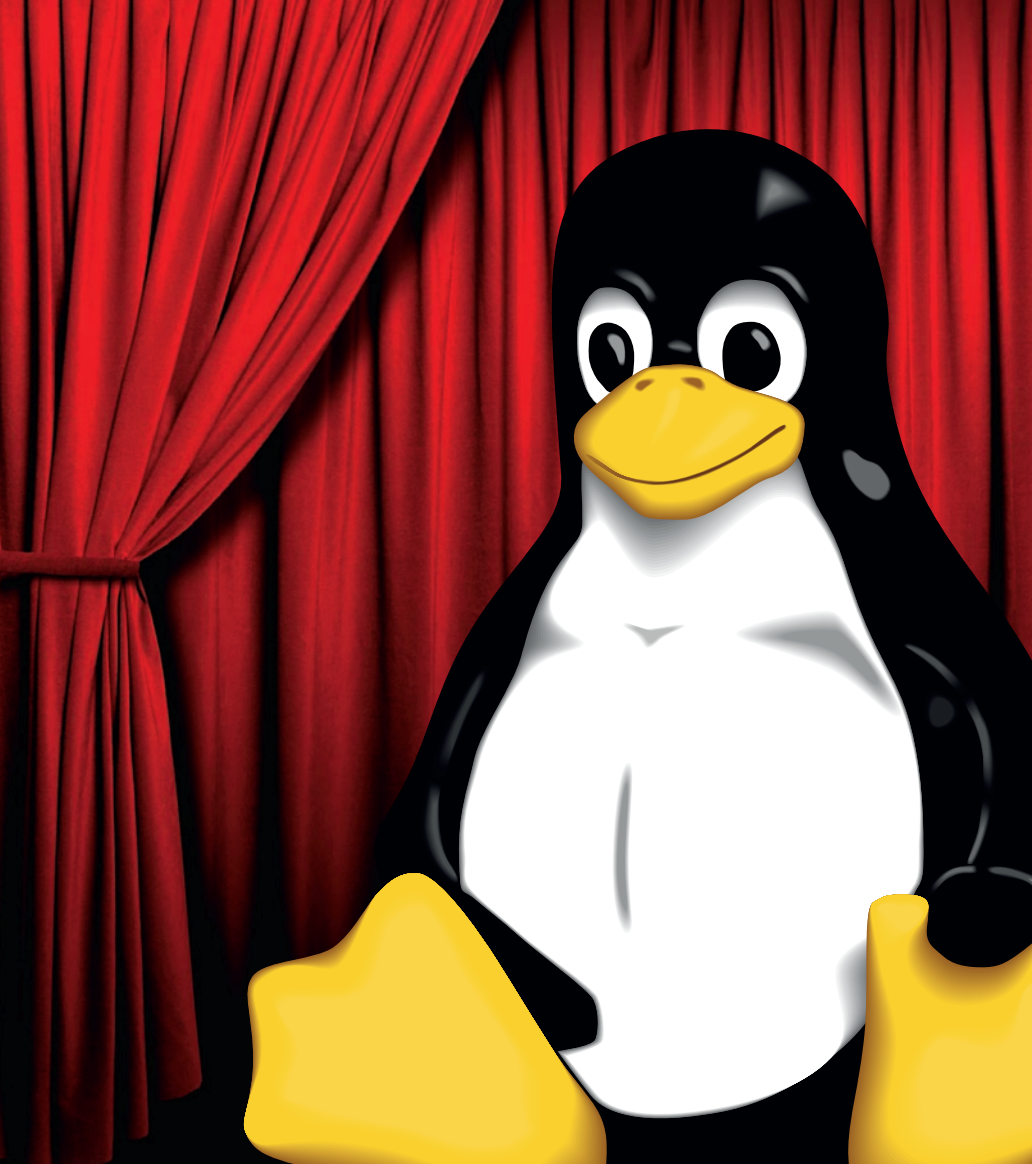
Mandriva Linux Powerpack 2009.1 Spring — 32- и 64-битные версии (2 DVD), а также печатное руководство ★ Mandriva Free 2009.1 Spring ★ Mandriva One 2009.1 Spring ★ Репозиторий Mandriva 2009 — бинарные пакеты для платформы x86 (4 DVD) ★ EduMandriva (1 DVD) — дополнительное ПО для образования ★ Академическая лицензия

www.mandriva.ru

Тел.: (812) 309-06-86, (499) 271-49-55
info@mandriva.ru

По этой программе учебное заведение получает:

- ★ свежие версии дистрибутива Mandriva Linux (дважды в год)
- ★ доступ к обновлениям системы
- ★ техническую поддержку



В сентябрьском номере

Убойные приложения!

Мы порыскали по SourceForge, чтобы добыть вам лучшие приложения для вашего Linux-компьютера.

Linux для широкой публики

Linux бесплатен, с работой справляется и не виснет – почему же мы до сих пор тратимся на Microsoft?

Построим облако с Eucalyptus

Обеспечьте разделение данных внутри своей организации посредством корма для коал. Это не просто медведи!

Содержание последующих выпусков может меняться без уведомления: вдруг мы заиграемся в теннис...

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 редактор

Елена Толстякова

Переводчики

Илья Аввакумов, Ольга Кокорева, Светлана Кривошеина, Валентин Развозжаев, Алексей Опарин, Елена Толстякова, Татьяна Цыганова

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 «Скай ЛТД»

196210, Санкт-Петербург, Вазетная ул., 11, корп. 2, лит. А
Тел.: (812) 677-98-03
Заказ 3826

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.com

Заместитель редактора Эндрю Грегори [Andrew Gregory]

agregory@futurenet.com

Редактор по продвижению и сообществу Майк Сондерс [Mike Saunders]

mike.saunders@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

Подготовка материалов Джоно Бэкон [Jono Bacon], Нейл Ботвик

[Neil Bothwick], Крис Браун [Chris Brown], Энди Ченнел [Andy Chappelle],

Алекс Кокс [Alex Cox], Эндрю Грегори [Andrew Gregory], Дамьен Макферран

[Damien McFerran], Боб Мосс [Bob Moss], Адам Оксфорд [Adam Oxford],

Джонатан Робертс [Jonathan Roberts], Маянк Шарма [Mayank Sharma],

Шаашанк Шарма [Shashank Sharma], Майк Сондерс [Mike Saunders],

Евгений Балдин, Семен Есилевский, Татьяна Казанцева, Тимур Мубаракшин,

Павел Протасов, Тихон Тарнавский, Андрей Ушаков, Алексей Федорчук

Художественные ассистенты Стейси Блэк [Stacey Black],

Ник Кокс [Nick Cox], Фил Хейкрафт [Phil Haycraft]

Иллюстрации Крис Хедли [Cris Hedley], Крис Винн [Chris Winn],

Ely Walton Illustrations, iStockPhoto

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 ЭМИ)

Тел./факс +7 (499) 271-49-54

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

LINUX – зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

«GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futurepic.com>

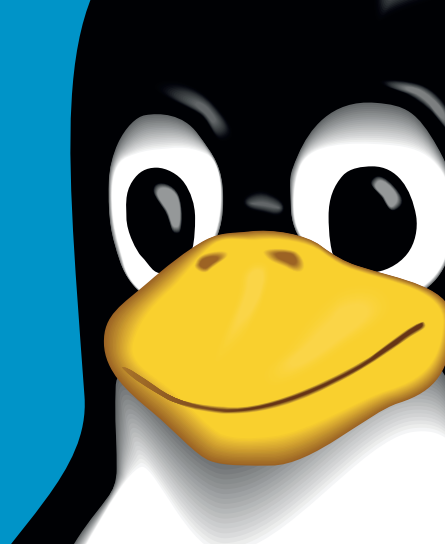


© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

Откажитесь от коммерческого ПО — сэкономьте годовой бюджет!



Операционная система GNU/Linux и свободное программное обеспечение помогут вам с минимальными затратами решить проблему лицензирования программного обеспечения, повысить безопасность и надежность вашей компьютерной сети.



**С нашей
помощью
вы сможете**

**Снизить затраты
на программное
обеспечение**



**Существенно
сократить время
на обслуживание сети
и устранение сбоев
в программном
обеспечении**



**Защитить сеть
и персональные
данные
с помощью оборудования
и дистрибутивов Linux,
сертифицированных
ФСТЭК**

ТИПОВЫЕ ПРОЕКТЫ:

- миграция рабочих станций и серверов с Microsoft Windows на GNU/Linux;
- установка 1С на серверах и рабочих станциях под управлением GNU/Linux;
- миграция с Microsoft Windows Active Directory на Mandriva Directory Server;
- миграция с Microsoft Exchange на Zimbra;
- внедрение интернет-телефонии на базе Asterisk;
- внедрение свободной CRM-системы SugarCRM;
- создание кластеров высокой доступности;
- реализация терминальных решений;
- создание порталов любой сложности на базе свободных CMS-систем — Joomla, Drupal, Plone;
- создание локализованных и кастомизированных версий операционных систем MeeGo, Android для OEM производителей и продавцов нетбуков, неттопов, планшетов, смартфонов и моноблоков.

СРЕДИ НАШИХ КЛИЕНТОВ:

- Минздравсоцразвития РФ;
- Министерство юстиции РФ;
- Правительство Московской области;
- Правительство Нижегородской области;
- администрация Черниговского района Приморского края;
- Министерство финансов республики Саха (Якутия);
- Владивостокский государственный университет экономики и сервиса;
- корпорация Intel;
- сеть магазинов «Компьютер-центр «Кей»;
- группа компаний «ИМАГ»;
- компания «Азбука мебели»;
- компания «Бестли — выставочные материалы».

Решите проблемы лицензирования ПО 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

ВЫДЕЛЕННЫЕ СЕРВЕРЫ

HETZNER ONLINE

ПРОСТО НЕОТРАЗИМЫ!

**СТОИМОСТЬ
УСТАНОВКИ
СНИЖЕНА!**

Все цены указаны без учёта НДС и регулируются условиями предоставления услуг компанией Hetzner Online AG. Цены могут быть изменены. Все права защищены соответствующими производителями.

HETZNER DEDICATED ROOT SERVER EQ 4

- Intel®Core™ i7-920 Quad-core с поддержкой технологии Hyper-Threading
- 8 GB DDR3 RAM
- 2 x 750 GB SATA-II HDD (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- Rescue система
- Установка из образов
- 100 GB для резервных копий
- Без минимального контракта
- Стоимость установки 1900 рублей

1900
рублей в месяц

HETZNER DEDICATED ROOT SERVER EQ 6

- Intel®Core™ i7-920 Quad-core с поддержкой технологии Hyper-Threading
- 12 GB DDR3 RAM
- 2 x 1500 GB SATA-II HDD (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- Rescue система
- Установка из образов
- 100 GB для резервных копий
- Без минимального контракта
- Стоимость установки 2700 рублей

2700
рублей в месяц

HETZNER DEDICATED ROOT SERVER EQ 8

- Intel®Core™ i7-920 Quad-core с поддержкой технологии Hyper-Threading
- 24 GB DDR3 RAM
- 2 x 1500 GB SATA-II HDD (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- Rescue система
- Установка из образов
- 100 GB для резервных копий
- Без минимального контракта
- Стоимость установки 3500 рублей

3500
рублей в месяц



Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выбирайте 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 Мбит/с. Стоимость аренды постоянного канала с пропускной способностью 100 Мбит/с составляет 290 рублей за каждый дополнительный ТБ.