

БЕСПЛАТНЫЙ DVD: Fedora 13 + Linux Mint 9 + Mepis

ВНУТРИ: 8-ГБ диск, полный дистрибутивов и программ

LINUX FORMAT



Fedora 13
Что в ней нового,
и почему это важно **с. 30**

Главное в мире Linux

Август 2010 № 8 (134)

В ЭТОМ НОМЕРЕ

Советы по Linux

» Наши фирменные учебники

Open Street Maps – примите участие

» Создайте свою карту мира вместе с другими

Восстановите поврежденные файлы

» Потеряли файл? Возьмите Par2

Обзоры: MythTV 0.23

» Больше функций, больше возможностей

Web-браузеры: кто лучший?

» Сравним Firefox 3.6 с Chrome 5



Дэвид Зигель
Самая большая возможность для Ubuntu – принести людям радость
О дизайне Ubuntu с. 28

Linux против iPhone

Android и MeeGo начинают и выигрывают



Плюс! Quickly
Быстрое создание приложений Python на с.68

GETTING THINGS GNOME!

Дела в порядке

» Управляем текучкой по Дэвиду Аллену

PERL

Рисуем графику

» Динамические изображения с помощью библиотеки GD

HARDCORE LINUX

Познакомьтесь с Mutt

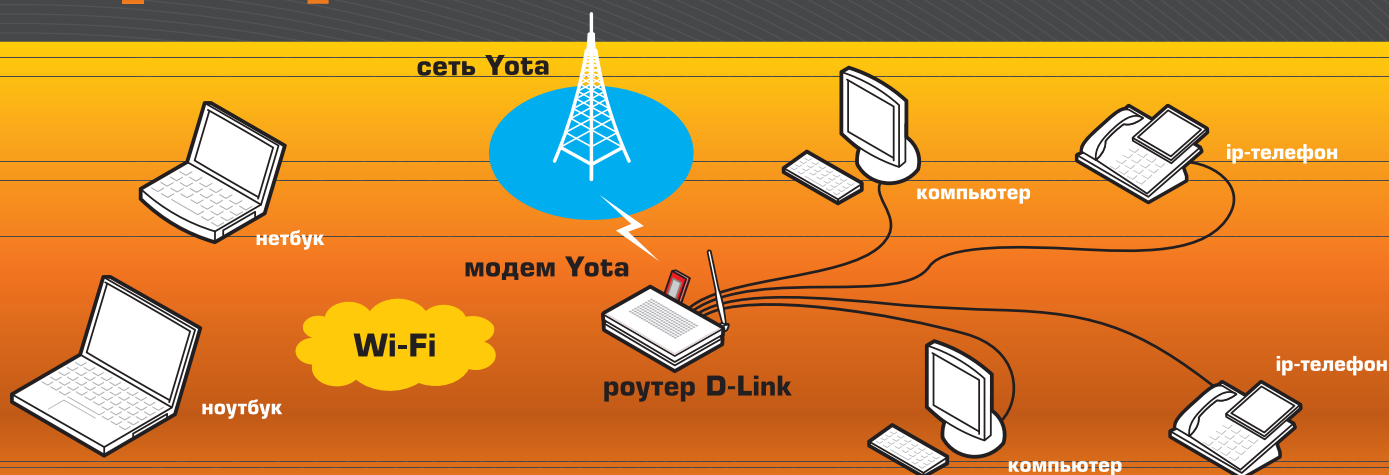
» Проверка почты, не отходя от терминала

Подписной индекс в каталоге агентства «Роспечать» 20882
Подписной индекс в каталоге «Пресса России» 87974
Подписной индекс в каталоге «Почта России» 16572



WiMaxStore

КОРПОРАТИВНЫМ КЛИЕНТАМ



Компания WiMaxStore рада предоставить вам комплекс услуг по организации доступа к безлимитному беспроводному высокоскоростному интернету Yota WiMAX 4G

- Тестирование скорости интернета (уровня сигнала Yota) в офисе клиента
- Доставка оборудования для организации интернета Yota 4G
- Подключение и настройка оборудования
- Техническая поддержка в течение одного года
- Разработка заказных решений для работы в сетях Yota 4G
- Организация VPN
- Обучение сотрудников клиента

Абонентская плата для юридических лиц, в зависимости от количества абонентов, составляет максимум

1400 руб./мес.

Стоимость указана с учетом НДС.

Интернет-решение организуется на базе роутеров D-Link DIR-320 или ASUS WL 500gP с модемом Samsung SWC-U200.

Оборудование, поставляемое компанией WiMaxStore, имеет все необходимые сертификаты.



 **wimaxstore**

САНКТ-ПЕТЕРБУРГ
+7 812 309-06-86

МОСКВА
+7 499 271-49-54

WWW.WIMAXSTORE.RU



Yota — разработчик и поставщик мобильных сервисов на базе самой современной технологии связи 4G (Mobile WiMAX). Yota — это первая в России сеть Mobile WiMAX (стандарт IEEE 802.16e-2005 на частотах 2,5—2,7 ГГц), развернутая в Москве и Санкт-Петербурге.

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

В этом номере мы сталкиваем лбами Linux и iPhone, а потому спросили наших гуру: что такого умеет Linux, что не по зубам детищу Apple?



Грэм Моррисон
Запустить *KAlum* – прекрасное приложение гениального автора. Ну, мне так говорили.



Майк Сондерс
Менять внешний вид. Потому что я хочу, чтобы мой телефон был похож на RISC OS, черт побери!



Нейл Ботвик
Как пользователь Gentoo, я рад перекомпилировать ядро ОС телефона, сучкая на автобусной остановке.



Эфраим Эрнандес-Мендоса
Работать с мышью и клавиатурой. И вичестером. И принтером. И 32-дюймовым монитором в придачу.



Эндрю Грегори
Благодаря виртуальным терминалам и *Screen*, я смогу говорить с несколькими людьми сразу. Наверное.



Знди Ченел
Синтезатор речи + режим психиатра *Emacs Meta + X* = я больше не одинок!



Дэвид Картрайт
В Linux можно запустить *Qeti*, а поверх него – систему всех времен и народов, 386BSD!



Энди Хадсон
Он защищает меня от полей, искажающих реальность вокруг продуктов Apple. Я... не... поддамс...



Ник Вейч
Открыв оболочку и набросав скрипт, я могу доставать Майка SMS-кой хоть раз в пять минут!



Сюзан Линтон
Чтобы пользоваться Linux, не надо постоянно говорить: «круто», «дизайн» и «инновационный».



Шашанк Шарма
В Linux я могу сделать DoS-ат... протестировать неудобные web-сайты, попивая чай в парке.



Боб Мосс
iPhone не даст вам позвонить Торвальдсу и сказать, что он нереально крут. Какая досада!



Сеть в кармане

» Нам говорят, что мобильность – одна из отличительных черт современного мира, как вообще в целом, так и в области информационных технологий. Наиболее смелые даже утверждают, что через какое-то время компьютеры в привычном смысле если не исчезнут, то существенно поубавятся в количестве: у вас всегда будет с собой маленькая коробочка, которую можно будет подключить к полноценному экрану (проектору) и клавиатуре, в изобилии развешанных по городу. Не такая уж широкая доступность столь старой вещи, как Интернет, заставляет усомниться в этом сценарии, хотя то, что значительная часть моих знакомых давно пересела с ПК на ноутбуки – это факт.

В этом смысле, я ретроград. У меня нет iPhone (и что меня больше всего изумляет – никогда не возникало желания его купить), и мой самый обычный сотовый телефон образца 2006 года редко занимается чем-то кроме приема и передачи звонков, сообщений, изредка – GPRS-трафика и фотографий. Мой текущий ноутбук с диагональю 15,6" мощнее моего настольного ПК, но я работаю преимущественно за последним.

И тем не менее, я с удовольствием слежу за развитием мобильных Linux-систем. Надеюсь, что и вы тоже: как показывает мой личный опыт, техническая оснащенность и техническое любопытство – не одно и то же. А раз так, открывайте стр. 20 и погружайтесь в сравнение ведущих закрытых и свободных платформ для мобильных устройств лицом к лицу. Вам не нравится Google? Посмотрите на обзор MeeGo на стр. 9. Ну а я, тем временем, пригляжу себе смартфон с Android или какой-нибудь планшет...

Валентин Синицын, 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

Нашими трудами внутри этого номера **LXF** всем найдется масса интересного.

Обзоры

CrossOver Games 9 8

Играть в Windows-игры под Linux безо всякой головной боли? Где мне подписаться?



» Отведите душу, поливая виртуальным свинцом виртуального противника.

MeeGo 1.0 9

Сотрудничество Intel и Nokia принесло свой первый плод.

KDevelop 4 10

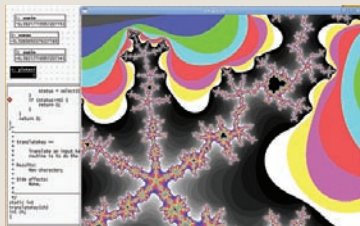
Переписанный с нуля, и все же функционально неполный. Это мы про KDE 4 или про KDevelop 4?

MythTV 0.23 11

Мощнейший программный мультимедиа-центр в мире по-прежнему может похвастаться мощнейшими трудностями в...

UndoDB 3.0 12

Отладка с перемещением назад по времени: удивительно, но это работает.



» Когда скорость написания кода достигает околосветовых величин, время обращается вспять.

Сравнение: Web-браузеры

Eriphany	16
Firefox	15
Flock	15
Google Chrome	18
Konqueror	17
Midori	16
Opera	17
SeaMonkey	18

Linux против iPhone

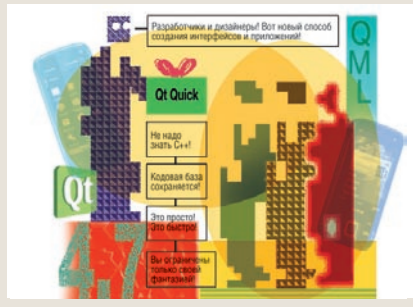
Битва начинается здесь! с. 20



Что за штука...

Qt Quick

Создавать приложения Qt стало просто, как никогда с.46



Fedora 13 с.30

Взлет за взлетом самого технически продвинутого дистрибутива в мире.



Люди говорят



« Еще многое надо сделать, но и это — уже дерзкий вызов привычным представлениям »

Дэвид Зигель про Ubuntu Unity с. 28

Новые дистрибутивы!

Диск с тройной загрузкой + вторая сторона

- » Fedora 13 Новейшие технологии мира Linux
- » Linux Mint 9 Ubuntu + дополнительные функции
- » Meris 8.5 Быстрый и дружелюбный к новичкам
- » Школьные дистрибутивы Попробуйте их в деле

Плюс: HaikuOS, MythTV и игры! **с. 100**

LXF DVD134

Ищите в этом номере...

Open Street Map 34

Примите участие в создании свободной карты мира, добавив на нее свой дом, любимую дорожку в парке или даже что-нибудь покрупнее.

Программируем периферию 38

Компьютер с Linux как центр умного дома – это вполне реально, особенно если знать, как правильно подключить к нему все эти датчики.

Vala и Genie 42

Едва ли мы испытываем нехватку языков программирования, так зачем создавать еще два? Или: как победить Microsoft .NET на его поле.

Пропустили номер?

Узнайте на с. 107, как получить его прямо сейчас!



Постоянные рубрики

- Новости** 4
События мира Linux глазами наших экспертов.
- Сравнение** 14
В котором Firefox получает пинок от Google Chrome. К сожалению.
- Интервью LXF** 28
Интерфейс ждет новая революция.
- Что за штука** 46
Qt Quick: почти как Ubuntu Quickly, но на Qt и без LY.
- Рубрика сисадмина** 48
Узнайте, как держать приложения под замком. Технологий для этого
- несколько, но выбор – это же здорово, верно?
- Ответы** 88
Проблемы Linux решены: от Grub до Dropbox.
- Hotpicks** 94
Лучшие в мире новинки свободного ПО.
- Диск Linux Format** 100
Двухсторонний DVD вновь под микроскопом.
- Пропустили номер? ...** 107
Закажите себе один из предыдущих выпусков журнала.

- Школа LXF** 108
Что ставить на школьные компьютеры к 1 сентября: обзор дистрибутивов.
- Через месяц** 112
LXF135 прекрасен и неповторим, как снежинка.

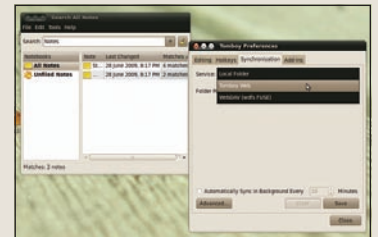


» А вы не забыли оформить подписку на LXF?

Учебники

Начинающим Облачный резерв 54

Узнайте, как сохранять свои файлы в облачном сервисе Ubuntu, для пущей надежности, и настройте горячие клавиши в KDE.



» Tomboy и сам неплохо хранит заметки, но почему бы не поделиться ими с друзьями?

Getting Things GNOME! Дела в порядке 58

Дело Аллена живет и процветает – справляйтесь с послепусковым авралом в GTG, даже если предпочитаете KDE.

par2 Чиним файлы 62

Немного побеспокоившись заранее, вы сэкономите кучу нервов потом – Par2 устранил повреждения в ваших файлах.

OLAP Многомерный анализ 64

Это сложная тема, но даже ее можно объяснить простыми словами, используя Qt и стандартные контейнеры.

Python Проект по-быстрому 68

Познакомьтесь с новым инструментом от Ubuntu и создайте полноценное приложение по восьми страницам.

Perl Выводим графику 76

Благодаря возможностям кросс-платформенной библиотеки GD, Perl может генерировать изображения на лету.

OpenOffice.org Calc Оптимизируем 80

Трюки, которые позволят вашей программе выполняться быстро: как универсальные, так и специфичные для StarBasic.

Hardcore Linux Почта из терминала 84

Thunderbird? Y! Evolution? Y! Истинные хакеры проверяют почту, не покидая терминала: в этом им помогает Mutt.

ГЛАВНОЕ Novell Pulse » Flash останется на YouTube » Новый OVI Store
» СКИФ-Аврора ЮУрГУ » Cuneiform 1.0 » Что волнует линуксоида?

Novell Pulse: Корпоративный Wave?



» Рубрику ведет
ЕВГЕНИЙ КРЕСТНИКОВ

С помпой дебютировав в прошлом году, Wave быстро исчез со страниц компьютерной прессы – но мы нашли повод его вернуть.

Сейчас уже очевидно, что будущее отрасли – за облачными сервисами, и многие корпорации рассматривают возможность перевода своего бизнеса в Сеть. Вскоре на этот рынок планирует выйти Novell – компанию трудно назвать новичком в области средств коллективной работы, но подобный онлайн-сервис она запускает впервые. По словам технического директора Novell СНГ Кирилла Степанова, Pulse будет доступен во второй половине 2010 года.

Идеологи проекта говорят о нем как о корпоративном средстве совместной работы: сотрудники компании имеют собственные публичные профили с текущими статусами и сведениями об областях знаний, в которых они ориентируются. Pulse позволяет обмениваться мгновенными сообщениями и совместно работать над документами. Кроме того, каждый пользователь имеет свой микроблог, может создавать группы и подписываться на обновления других пользователей и групп. Электронные сообщения из различных систем накапливаются в специальных унифицированных ящиках, а все правки и комментарии видны в реальном масштабе времени: если в сообщении содержится документ, понятно, кто и какие изменения в него вносит – файл автоматически синхронизируется с сервером и обновляется у всех пользователей. При этом в сервисе реализован контроль версий, и можно увидеть состояние документа в любой момент его жизненного цикла. Новый сервис очень похож на Google Wave (и может интегрироваться с ним через Wave Federated Protocol), но это собственная разработка компании, использующая написанный на Scala инструментальный Lift 1.0. Разработчик Pulse Дэниэл Спивак



» **Дизайн Novell Pulse, похоже, охраняется не меньше, чем когда-то – внешний вид Vista, но в Сети можно найти видеоролики.**

[Daniel Spiewak] заявил: «Поддержка инструментарием технологии Comet упростила архитектурные решения. Мы используем Comet, и выбор Lift принес проекту реальную пользу. Интерфейс web-приложения полностью реализован на Scala. Большая часть движка написана на Java, но для сложных функций, таких как рендеринг схемы содержимого Google Wave в XHTML, мы также используем Scala».

Главная особенность Pulse, отличающая его от сервиса Google – возможность интеграции в корпоративную систему авторизации и развития средства управления правами доступа, которых еще нет в Wave. В Novell СНГ пока ничего не говорят о модели лицензирования продукта, а многие функциональные возможности Pulse (например, совместное редактирование документов) еще не доступны для тестирования. Кроме того, неизвестно, будет ли Pulse доступен заказчикам только в виде сервиса (SaaS), или появится его локальная версия. В дальней перспективе возможно объединение Pulse с Teaming – другим продуктом Novell для коллективной работы. Что касается Wave – уже известны планы Google по включению продукта в состав Google Apps. Можно ожидать, что «корпоративный» функционал в него добавят, и у Pulse появится конкурент. Но пока его полных аналогов на рынке нет. Тем не менее, тенденция миграции крупнейших разработчиков ПО в Сеть видна вооруженным взглядом: не только Google, но и ряд других компаний (включая Microsoft) движутся в этом направлении.

Что будет? Всероссийская конференция «Свободное программное обеспечение-2010»

» 26 и 27 октября 2010 года, Санкт-Петербург

Конференция пройдет в Санкт-Петербургском государственном политехническом университете. Организаторами конференции выступают Некоммерческое партнерство содействия развитию свободного программного обеспечения (РАСПО) и Санкт-Петербургский государственный политехнический университет (Факультет технической кибернетики СПбГПУ), при поддержке Министерства связи и массовых коммуникаций РФ, Министерства образования и науки РФ.

Основной целью конференции является обсуждение возможностей СПО как инструмента для развития инфор-

мационных технологий и перспектив по продвижению СПО в различные сферы деятельности: в структуры государственного управления, в бизнес, индустрию разработки ПО, научные и образовательные учреждения.

В ходе конференции состоятся пленарные доклады известных ученых и специалистов в области СПО, секционные доклады, пройдет круглый стол с участием представителей органов государственной власти, науки, образования и промышленности, заинтересованных в развитии отрасли программного обеспечения в России и рассматривающих модель СПО как важный инструмент такого развития.

<http://www.raspo.ru/spo2010>

YouTube не откажется от Adobe Flash

Технологии, включенные в HTML 5, оказались недостаточно гибкими для популярного видеосервиса, утверждают эксперты.

В блоге для разработчиков YouTube Джон Хардинг [John Harding] официально заявил, что сервис пока не будет отказываться от Adobe Flash. По сообщению г-на Хардинга, тэг <video> не обеспечивает всех необходимых сервису функций. Эксперт говорил об отсутствии единого формата из-за проблем лицензирования H.264 (впрочем, тут же упомянув WebM). Неясно, какое отношение имеет формат контента к способу его воспроизведения (тем более что Adobe планирует включить поддержку WebM в свои продукты), но другие причины выглядят более убедительно.

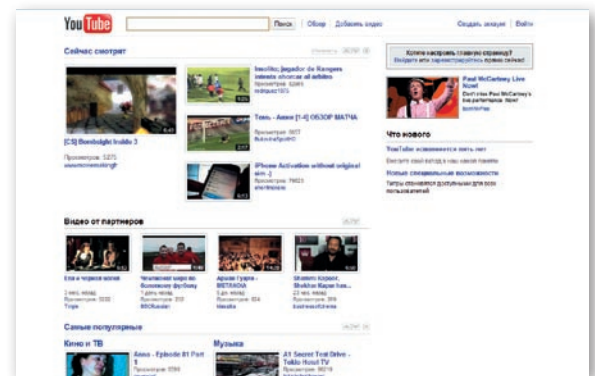
В частности, речь шла о плохой управляемости: Adobe Flash может управлять проигрыванием через ActionScript, в то время как тэг <video> не дает пользователю возможности получить определенную часть ролика. В блоге говорится, что проблемы с буферизацией сейчас решаются, и корпорация Google вносит в работу свой посильный вклад. Есть трудности и с полноэкранным воспроизведением: в современных браузерах поддержива-

ется такой режим, но инициировать его из JavaScript или развернуть определенную часть страницы (например, плеер) они не позволяют. Задачу должны решить разработчики браузеров, но явные шаги в этом направлении делает только WebKit. Другой камень преткновения – авторские права: многие владельцы контента требуют от YouTube использовать защищенные протоколы, такие как RTMPE Flash Platform. Последней важной причиной стало отсутствие в HTML 5 поддержки работы с камерой и микрофоном. Представитель Google говорил и о плохой встраиваемости HTML 5 – по его мнению, только Flash Player может объединить контент и код в некий пакет, позволяющий безопасно встраивать содержание сторонних сайтов. Мне эта причина кажется несколько надуманной, хотя спорить со знатоками web-технологий я бы не взялся.

Претензии к новым стандартам высказаны довольно серьезные (и большей частью обоснованные), хотя главную причину г-н Хардинг назвать постеснялся: стандарты HTML 5 не позволяют демон-

стрировать сопутствующую рекламу (ролики на тестовой версии YouTube ее не содержат). Наверное, корпорации стоило бы активнее использовать свободные форматы, не относящиеся к HTML 5 (тот же SVG в сочетании с тэгом <video> позволяет создавать очень интересные решения), но от Flash отказываться пока рано. Активное продвижение свободной замены Flash началось, и это уже хорошо, но работа предстоит еще немалая.

➤ Судя по всему, избавиться от Flash в популярном видеосервисе удастся еще не скоро.



Nokia меняет политику использования Ovi Store

Распространять приложения через официальный магазин компании теперь станет проще.

В корпорации Nokia решили упростить жизнь разработчикам мобильных приложений. Для загрузки доступен Nokia Qt Development Kit (SDK) 1.0, позволяющий создавать кросс-платформенные приложения для Symbian и Maemo (в скором времени будет поддерживаться и MeeGo). В июле онлайн-магазин компании начал распространение Qt-приложений: пока они доступны для Nokia N8 (здесь уже есть библиотеки Qt 4.6) и всех устройств на платформе Symbian S60, начиная с версии 3.1, для которых в приложение нужно включить Smart Installer, входящий в состав Qt Development Kit. Он запускается при установке, проверяет зависимости и докачивает необходимые пакеты. Кроме того, сейчас любой индивидуальный разра-

ботчик может получить статус Ovi Publisher и распространять свои приложения через Ovi Store – правда, за это придется заплатить разовый взнос в размере €50. Запущена программа бесплатной сертифика-

«В корпорации Nokia решили упростить жизнь разработчикам.»

ции приложений Qt, Symbian C++, Flash Lite и Web Runtime при отправке их в Ovi Store. Воспользоваться ей может любой – Publisher ID не требуется. Процедура сертификации в рамках Public beta of Nokia signing должна занимать до двух недель,

а успешно прошедшие ее программы будут распространяться только через Ovi Store – получить обратно подписанный sis-файл нельзя. И наконец, разработчики смогут загружать в Ovi Store Symbian-приложения для Nokia N97 mini и Nokia X6, а также приложения для Maemo (Nokia N900). Еще есть информация о скором переходе Qt на открытую модель управления, и ходят слухи, что Nokia N8 станет последним флагманом N-серии на Symbian, а следующие коммуникаторы будут выпускаться только с MeeGo. Впрочем, подобные слухи ходили перед выпуском N900.

Новости это хорошие, но не стоит забывать, что компания сейчас в непростом положении и может утратить статус мирового производителя № 1. Поживем – увидим...

СКИФ-Аврора ЮУрГУ: Еще один суперкомпьютер

Вычислительный комплекс сейчас занимает восьмую строчку в Top-50 самых мощных суперкомпьютеров СНГ.

На совместной пресс-конференции, организованной университетом ЮУрГУ, Институтом программных систем РАН, компанией «РСК СКИФ» и корпорацией Intel была проведена демонстрация работы суперкомпьютера «СКИФ-Аврора ЮУрГУ».

Комплекс использует систему жидкостного охлаждения и содержит 256 «лезвийных» модулей с двумя процессорами Intel Xeon X5570 (4 ядра с тактовой частотой 2,93 ГГц). Пиковая производительность установки – 24 ТФлоп, а производительность на тесте LINPACK – 21,84 ТФлоп. Разработанный в рамках программы «СКИФ-GRID» Союзного государства комплекс вошел в Top-50 самых мощных суперкомпьютеров СНГ и занял восьмое место в списке. Его реализацией занималась компания «РСК СКИФ» совместно с дру-

гими исполнителями программы, при поддержке западных партнеров. На суперкомпьютере используется операционная система ALT Linux SKIF.

То, что в России и Беларуси могут строить суперкомпьютеры – это хорошо. Но читая о запуске очередного вузовского вычислительного комплекса, не могу удержаться от вопроса: какими задачами загружают эти мощности? Чем сейчас занимаются, например, комплексы, построенные в России корпорацией IBM? Если верить циркулирующим слухам, задач для таких «монстров» пока нет, и значительная часть их ресурсов простаивает. Стоит задуматься о построении облачных сервисов, которые позволят и другим учебным заведениям получить доступ к серьезным вычислительным мощностям.

Cuneiform 1.0: Лучшее – враг хорошего

В Linux наконец-то появилась свободная система оптического распознавания образов.

Вышла первая стабильная версия OCR *Cuneiform* для платформы Linux. Продукт распространяется под упрощенной лицензией BSD и основан на исходных текстах, открытых компанией Cognitive Technologies в 2008 году. Система распознает полиграфические и машинописные гарнитуры, распечатанные на принтерах, а также машинописные тексты, факсы и ксерокопии. Она поддерживает более 20 языков, в том числе русский и украинский. Результат экспортируется в HTML, hOCR, RTF или PDF; при этом сохраняются особенности начертания исходного текста. В составе продукта – только консольные утилиты, но существуют

и две графических оболочки: *YAGF* нашего постоянного автора Андрея Боровского (см. стр. 38) и *Cuneiform-Qt* Андрея Черепанова из ALT Linux.

Пользователи подтверждают хорошее качество распознавания и сожалеют об отсутствии возможности сохранения исходного форматирования и распознавания таблиц. Эту проблему пытаются решить в *Tesseract/Octopus* (отдельный проект), но все иностранные разработки имеют серьезные проблемы с русскоязычным текстом. Несмотря на все недостатки, *Cuneiform* остается самой «продвинутой» OCR для русскоязычных пользователей, и мы надеемся, что проект будет развиваться.

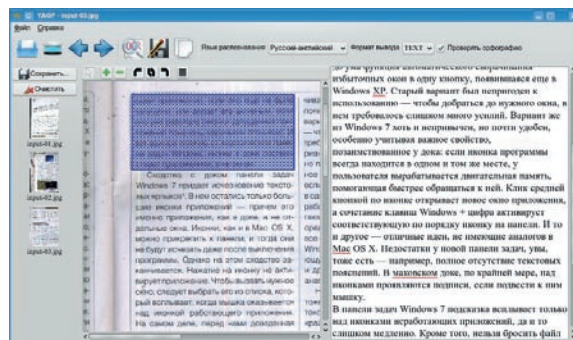
Linux.Org.Ru: Топ-5

Что интересует линуксоида? Попробуем разобраться, взяв в качестве индикатора 5 самых «горячих» тем Linux.org.ru.

На пятом месте в списке (811 комментариев) – выход Opera 10.60 для Linux: сказывается высокая популярность браузера в странах СНГ. На четвертое взлетел новый инсталлятор Ubuntu 10.10: тема опубликована 13 июля, а 15-го комментариев стало 845. Третья позиция показывает, что сообществу близки проблемы программирования – тема форума «В чем заключается революционность перехода от функциональности к OOP?» собрала 924 комментария.

Очень интересна вторая ступенька пьедестала – на нее возшла Ирина Матюшонко со статьей «Россия. Далеко ли до мейнстрима?: Часть 1. Сферический Linux в российском вакууме». Материал оценили (не всегда положительно), и обсуждение собрало от 1046 до 1070 комментариев: автору удалось наладить диалог с читателями. Видимо, это не понравилось модераторам, и часть комментариев была стерта. Ирина говорит о ситуации так: «Удаление комментариев автора статьи в ее (статье) бурном обсуждении – парадокс милый, но, возможно, небезобидный. Небезобидный тем, что модератор популярного OpenSource-ресурса «вытирает» (хочется верить, по недоразумению) все официальные предложения о сотрудничестве – и популяризации СПО в крупных СМИ».

А на первом месте – обсуждение причин, отпугивающих новичков от Ubuntu. Опубликованные командой Canonical Design Team результаты исследования породили 26 страниц из 1295 комментариев. Новость удерживает пальму первенства, и Ubuntu стал актуальнейшей темой месяца.



➤ Cuneiform 1.0 – важный шаг вперед в области свободных проектов для оптического распознавания текстов.



Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
Его слабости — mass storage, разметка диска и файловые системы.

Памяти советской геологии

Не так давно образовался сам собой такой проект — сайт памяти советской геологии, <http://sovgeology.su>. Это — воспоминания о событиях тех лет и о людях, создававших минерально-сырьевую базу нашей страны. Ту самую, на которой она живёт и по сей день. Постараемся вспомнить всех — список персоналий уже сейчас очень обширный и будет пополняться по мере сил и возможностей.

Какое отношение это имеет к Linux? Да только то, что весь проект выполняется на Linux-машинах. Вот и дошла эта ОС до жизни такой, что в ней можно просто заниматься самыми разными практическими делами. В том числе, и геологическими. Не напрягаясь уже вопросами перекомпиляции ядра, прикручивания кириллицы или настройки X. Иначе говоря, началась нормальная цивилизованная жизнь. Не об этом ли мы мечтали долгие предшествующие годы?

А ещё — Open Source и геология на самом деле очень родственные явления, ибо основаны на одних и тех же принципах: свободе распространения базовой информации с целью построения на ней конечных решений.

К соучастию в проекте приглашаются все, кто в теме, кто знает и помнит...

alv@posix.ru

Сегодня мы рассматриваем:

- CrossOver Games** 8
Вероятно, это последний выпуск коммерческой среды для запуска Windows-игр в Linux до выхода родного релиза Steam. Это значит, что он должен быть хорош, или он будет потерян.
- MeeGo 1.0** 9
Будущее мобильного Linux заставляет Ubuntu побороться за место под солнцем на рынке нетбуков, и возможно, вы скоро увидите его на телефонах от Nokia.
- KDevelop 4** 10
Сейчас самая пора конкурировать со средой разработки XCode 4 от Apple — удастся ли KDevelop 4 привлечь в стан KDE больше независимых разработчиков?
- MythTV 0.23** 11
Не беспокойтесь — в этот раз мы не будем жаловаться на то, как сложно установить MythTV. А на сэкономленном месте расскажем, как нам понравился новенький видеомодуль.
- UndoDB** 12
Мы никогда не делаем ошибок в коде, который пишем, и поэтому были вынуждены подойти к обзору этого инструмента чисто умозрительно. И он прекрасен.



» После всех красивых слов, результат совместных трудов Intel и Nokia наконец-то можно пощупать на зуб.



» Если уж вам повезло добиться работоспособной настройки MythTV, можете ни в чем себе не отказывать.

Наш вердикт: Пояснение

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



Выдающиеся решения 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, но если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчики: Google

Сайт: <http://earth.google.com>

Цена: Бесплатно по закрытой лицензии

Функциональность 10/10

Производительность 9/10

Простота использования 9/10

Оправданность цены 9/10

» Если весь мир — сцена, то Google Earth — театр. Простая в использовании, захватывающая и ободряющая практическая программа.

Рейтинг 9/10

CrossOver Games

Король Windows-совместимости CodeWeavers обновил игровую часть своего ассортимента. Энди Ченнелу пришлось потратить пару дней...

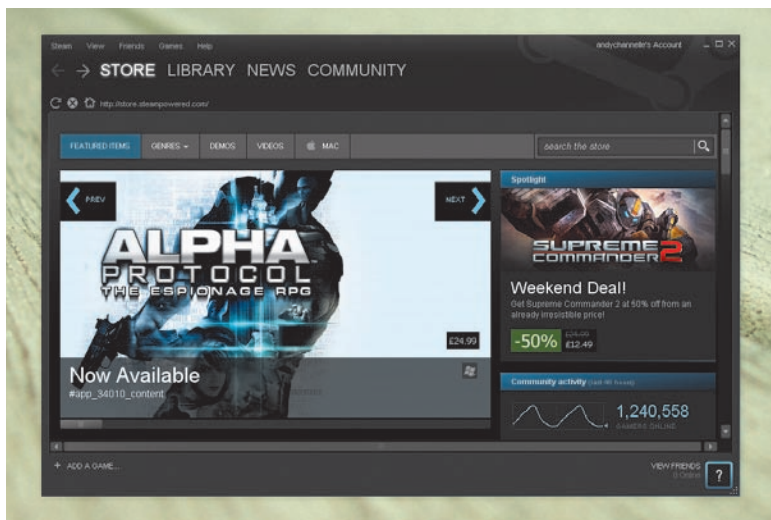
Вкратце

» Слой совместимости с Windows с игровым уклоном. См. также Wine, Cedega и CrossOver Office.

В основе *CrossOver Games* лежит *Wine* – слой совместимости, позволяющий работать с Windows-ПО на Linux, не оплачивая платформу Microsoft. Отличие от *Wine* состоит в том, что разработчики *Crossover Games* блестяще обеспечили работу с играми (есть и другая версия ПО от CodeWeavers – «серьезная», под названием *CrossOver Office*).

Подобно обновлённой офисной версии, *CrossOver Games* сменил внешность, ради гармонии с рабочим столом Gnome; улучшились система установки игр и управление капсулами. Последнее обновление жизненно необходимо, ведь именно программные капсулы позволяют подгонять имитацию среды Windows под особенности отдельных игр.

CodeWeavers делит игры мира Windows на «поддерживаемые» и «неподдерживаемые», причём в первом списке представлены все бренды от *Half Life* и *World of Warcraft* до новейших *Spore* и *Star Trek Online* (последняя – через *Community Support*). По быстрдействию игры ныне поч-



» Новая версия более уверенно работает с игровой службой Steam.

ПО устанавливается прямо из интерфейса *CrossOver Games* и получает новый пункт меню для запуска.

Техно-шок

После создания учётной записи Steam новый интерфейс наполняется демо-версиями, видео высокой чёткости и обзорами игр. В который уже раз напоминаем: не спешите раскошелиться, почитайте форум интересующей вас игры и загрузите демо-версию.

Стоит помнить и о том, что, несмотря на игровую направленность, *CrossOver Games* может работать с офисными приложениями, включая *MS Office*, ранние версии *Photoshop* и другие, поддерживаемые *CrossOver Office*, хотя эта возможность особо не афишируется.

ПО устанавливается в параллель *CrossOver Office* и версии *Wine*, применяемой для работы с *Picasa* от Google. Естественно, всё это дублирование требует больше ресурсов, чем применение разделяемых библиотек – зато стабильность на высоте. Если зависает *Picasa*, все остальные *Wine*-приложения остаются на вахте.

CrossOver Games – достойное обновление парка CodeWeavers, и если объект ваших вожделений входит в список поддерживаемых игр, приобрести программу определённо стоит. Неофициальную, но надёжную поддержку офисного ПО

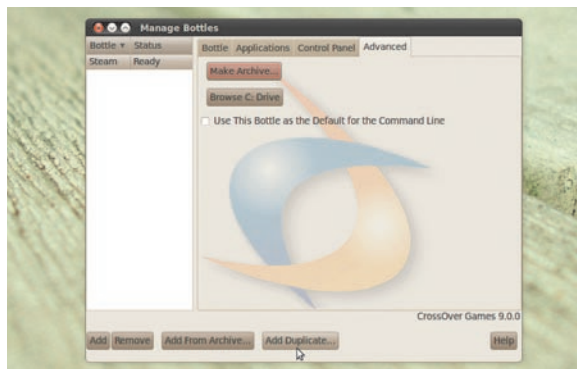
можно считать бонусом – то есть при наличии Windows-раздела можно по ходу, не перезагружаясь, поработать с документами. Функция архивирования (возможность «упаковать» настроенную капсулу для переноса на другую машину) тоже работает нормально.

Если во время игры работал *Firefox*, отмечались дефекты звука: вероятно, это результат «монополизма» браузера в использовании звуковой карты. Были затруднения и в настройке прокси через графический интерфейс. Но эта головная боль офисных игроков в домашней обстановке встречается редко, и лечится экскурсией в командную строку. **LXF**

«CrossOver Games – достойное обновление парка CodeWeavers.»

ти не отстают от оригиналов; в частности, много труда вложено в *World of Warcraft*.

По времени выпуск версии 9 почти совпал с обновлением пользовательского интерфейса на канале Steam, распределяющем игры от Valve, но программисты CodeWeavers успели и тут. Обновлённое



» Управление капсулами переделано: обновлены графика и поддержка архивирования и переноса на другие машины.

LINUX FORMAT **Вердикт**

Crossover Games 9.0

Разработчик: CodeWeavers
 Сайт: www.codeweavers.com
 Цена: \$39,95

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

» Гораздо дешевле лицензии Windows; главное, чтобы ваша игра входила в список поддерживаемых.

Рейтинг 8/10

MeeGo 1.0

Грэм Моррисон недоумевает, что вышло из MeeGo – продукта скрещивания Moblin и Maemo: сумма обеих составляющих или аннигиляция.

Вкратце

» Maemo + Moblin = MeeGo, платформа для мобильных компьютеров от Nokia и Intel. Среди альтернатив – Android и детище Canonical, UNR для нетбуков.

МееGo – неразрывная комбинация платформы Moblin, созданной Intel, и системы Maemo от Nokia. Обе ОС изначально создавались для разных сегментов мобильного рынка. Moblin предназначен для нетбуков: графический интерфейс на основе Clutter специально разработан для экранов с разрешением 1024×600. Претензии Maemo гораздо скромнее: это система для смартфона со стилусом и экранной клавиатурой. Если вам непонятно, на каком участке эти платформы пересеклись с образованием MeeGo – то вы не одиноки.

Мы установили MeeGo рядом с Ubuntu UNR, причём на последней стадии установщик завис, записывая загрузчик на диск. Пришлось перезагрузиться в UNR и набрать **grub-update**. (такого не случилось бы, будь MeeGo единственной ОС.) Стандартная файловая система MeeGo – супер-передовая btrfs; решение спорное. Btrfs обладает несомненными достоинствами для мобильных уст-

«Холодная» загрузка и выход из спячки быстрее, чем в UNR.»

ройств, включая прочную сохранность данных и фоновую дефрагментацию – но не рано ли рекомендовать её для широкого применения? Справедливости ради заметим, что, пока мы работали с MeeGo, затруднений с файловой системой не было – исключая интенсивную разрядку батарей при записи файлов.



» Если вы считали Synaptic скучным пакетным менеджером – значит, не пробовали нищий Garage.



» Несмотря на новые цвета и темы, трудно разглядеть в MeeGo 1.0 существенные новшества – хотя многое, например, Qt 4.6, скрыто за сценой.

После загрузки в MeeGo предлагается создать учётную запись пользователя с возможностью шифрования домашнего каталога. Затем через пару секунд появляется рабочий стол MeeGo.

Оборудование определилось неплохо: почти все компоненты нашего Samsung NC10 были зарегистрированы и учтены, включая графический ускоритель и двухпальцевую прокрутку на сенсорной площадке. Единственный недостаток – пропущена функция изменения яркости экрана с помощью клавиатуры (UNR 10.04 здесь осечки не дает).

Скорость «холодной» загрузки и выхода из спячки по сравнению с UNR заметно выше – от подачи питания до полноценного рабочего стола с активным беспроводным подключением проходит всего 22,3 секунды, а выход из спящего режима занимает всего пять секунд.

Мигмом сбегу

Хотя MeeGo отважно позиционируется как средство связи с «облачными» приложениями, мы не нашли инструментов автоматической привязки к онлайн-календарю или электронной почте; нет ничего нового и в области просмотра твитов и обновлений с Facebook с экрана MyZone. Изрядно досаждают мелкие неполадки вроде несвоевременного обновления статуса.

Поскольку это Linux, всегда существует возможность установить новые пакеты – есть такое средство и у MeeGo 1.0 (Garage называется). Нынче трудно успе-

дить за названиями менеджеров приложений: все вразумительные имена уже разобрали.

Garage – менеджер пакетов в MeeGo – по умолчанию разочаровывает жалкими 16-ю позициями. Кроме того, при запуске некоторых программ (например, игр – клонов Klondike) в полноэкранном режиме пропадает доступ к панели быстрого запуска, что делает усовершенствования несколько избыточными.

Вот так мы оцениваем результат совместной работы Nokia и Intel. Конечно, многое скрыто от глаз – например Qt 4.6 в комплекте разработчика MeeGo. И мы знаем, что Intel плотно работает над поддержкой сенсорных устройств в версии 1.1. Но версия 1.0 оставляет желать лучшего. LXF

LINUX FORMAT Вердикт

MeeGo 1.0

Разработчик: Intel и Nokia
Сайт: <http://meego.com>
Цена: Бесплатно на условиях GPL

Функциональность	5/10
Производительность	9/10
Простота использования	7/10
Документация	6/10

» Неплохой выбор для скромного нетбука, но рядом с Android и iPad выглядит бледно.

Рейтинг 6/10

KDevelop 4



Полностью переписанный для KDE 4, *KDevelop* снова с нами и готов сбросить с пьедестала *Qt Creator*. Не так ли, Грэм Моррисон?

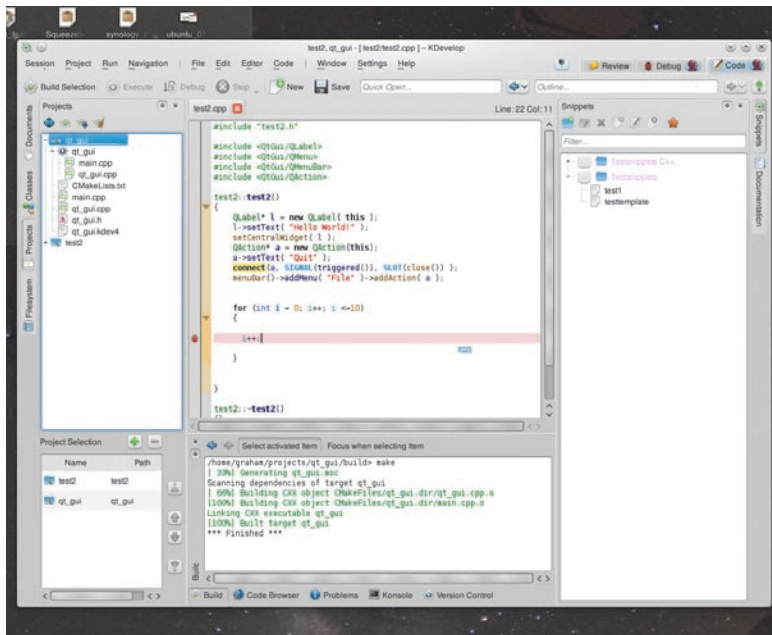
Вкратце

» Среда программирования для KDE и Qt с использованием языка C++. См. также *Qt Creator* и *Eclipse* или *Glade*.

Это второй случай полного переписывания стандартной среды разработки KDE за ее 11-летнюю историю. Первое было вызвано крупными изменениями *Qt 3* и *KDE 3* в 2004 году; текущее отражает переход обоих компонентов к четвертому поколению.

Первое впечатление от *KDevelop 4.0* ничуть не изменился. Философия графического интерфейса осталась прежней: вкладки и панели «растут» из левой, правой и нижней границ окна, а новая версия получила ещё три панели сверху, и теперь все четыре границы заняты выдвижными «карманами». Кроме того, по сравнению с прежней версией очень мало шаблонов для быстрого начала работы – серьёзный недостаток, особенно для новичка.

Но, преодолев эти трудности, обнаруживаешь реальные усовершенствования. Очень хороша интеграция с API KDE и *Qt*. Набор кода сопровождается подсказками из документации, а если навести курсор на ключевое слово, можно увидеть тек-



» Найдите десять различий: несмотря на полное обновление кода под KDE 4, *KDevelop 4* почти неотличим от предшественника.

и предоставляет выбирать из выпадающего списка вместо обращения к документации.

Про хорошее

Неплохо действует и семантическая подсветка, в которой фоновое знание API и языка программирования используется для обнаружения ошибок и поиска решений. Например, если создать цикл *for* с неинициализированным параметром, *KDevelop* уведомит об ошибке и предложит объявить либо локальную на уровне функции или цикла, либо закрытую переменную.

Как и в прежней версии, отлаживать приложение можно прямо в главном окне, устанавливая контрольные точки и выверяя участки кода без обращения к командной строке. Но разработчики лишены возможности использовать графический интерфейс для обратной отладки *GDB 7*, что могло бы дать *KDevelop* бесспорное превосходство над соперниками. Нет и обратной совместимости: поддержка *Qmake* и *Automake* прекращена ради новинки – *CMake*, которая поддерживается и *Qt 4*, и *KDE 4*. Потеряна и поддержка в *KDevelop* других языков программирования, хотя работа в этом

направлении идёт. Официально выпущены только модули для C++ и PHP, но в разработке находятся Ruby, Java и CSS.

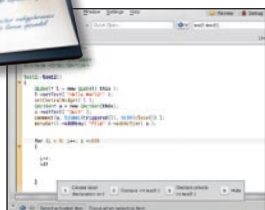
В этом вся беда *KDevelop 4*. Это отличная платформа, интегрированная в API, с завершением кода и подсветкой синтаксиса. Но если функциональность новой версии даже близко не стоит с возможностями прежнего инструмента, нужна очень веская причина для обновления. В случае с *KDevelop 4* такой причины не просматривается. **LXF**

«Завершение кода ладит с сигнально-слотовым механизмом Qt.»

стовое описание его функций. Завершение кода отлично ладит с сигнально-слотовым механизмом *Qt*: *KDevelop* «знает», каким слотам какие сигналы соответствуют,

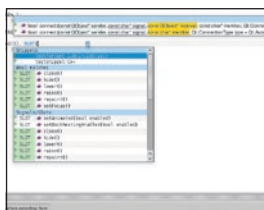


Свойства навскидку



Автопоправки

KDevelop замечает распространённые ошибки кодирования и даже предлагает способы их устранения – например, в случае неправильного цикла *for*.



Завершение кода

Удерживайте сочетание клавиш, чтобы получить перечень возможных вариантов завершения текущей строки.

LINUX FORMAT Вердикт

KDevelop 4

Разработчик: Команда KDevelop
Сайт: www.kdevelop.org
Цена: Бесплатно на условиях GPL

Функциональность	7/10
Производительность	7/10
Простота использования	4/10
Документация	6/10

» Объединив *KDevelop* и *Qt Creator*, мы получили бы потрясающую среду программирования.

Рейтинг **6/10**

MythTV 0.23



Не сломленный прежними схватками с MythTV, Грэм Моррисон в который уже раз пытается приручить упрямаца.

Вкратце

» Мощный цифровой видеорекодер, умеющий всё на свете. Альтернативы: бесплатный *LinuxMCE* и коммерческий *SageTV*.

Э то первое обновление *MythTV*, свободного приложения для записи и воспроизведения телепрограмм, с момента перехода на новый цикл разработки. Вместо двухлетнего периода программа будет обновляться каждые шесть месяцев. Хотя при взгляде на перечень новых функций этого не скажешь: он по-прежнему необъятен. Версия 0.23 получила новые модули расширения, новую тему, кучу внутренних улучшений и переработанный звуковой движок.

На сей раз мы не станем жаловаться на мучительную процедуру установки и настройки. Довольно мы это делали, а она упорно остается испытанием на прочность: ошибка – и ты готов рехнуться, успех – получаешь лучший в мире цифровой видеомаягнитофон, который за деньги не купить. Лучше сосредоточимся на позитиве. Традиционно для *MythTV*, после установки и запуска сразу бросается в глаза

«Изюминка данной версии – новый модуль MythNetvision.»

новая тема. Сейчас она называется *Arclight* и выглядит довольно мило и современно: графика в синих тонах элегантно размыта. Сразу после выхода новой версии в списке рассылки *MythTV* появилось несколько новых сообщений о дефекте новой темы (пикселизированное пятно справа). Но это только фоновое изображение; его легко заменить другим.

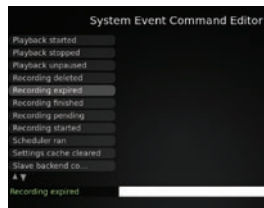


Свойства навскидку



Arclight

Новая тема быстро завоевала наши симпатии: на широком экране смотрится замечательно.



Диспетчер событий

Теперь скрипт можно «привязать» едва ли не к любому событию *MythTV*.



» Забудьте о широкоэвещательном ТВ: будущее за интернет-трансляцией. Интересно, как на этом можно заработать?

Изюминка данной версии – новый модуль *MythNetvision*. Его можно вызвать из главного меню выбором команды Media Library [Медиа-библиотека], а затем Search For Internet Video [Поиск видео в Интернете] либо Browse Internet Video [Интернет-видеосёрфинг]. Как и следовало предположить, он воспроизводит видеофайлы из Интернета без помощи браузера. Это напоминает интерфейс устройств типа *Popcorn Hour* (см. **LXF116**) и грядущий сервис *Google TV*.

Ловим сеть в Сети

В первом варианте вводится поисковый запрос, а затем выбирается служба из списка, где значатся *MTV*, *YouTube*, *Vimeo*, *BlipTV* и *DailyMotion*. Результаты отображаются в правой части окна, а нужный пункт можно выбрать пультом дистанционного управления. Воспроизведение напоминает модуль *MythVideo* и чаще всего полноэкранное. Некоторые видеофайлы воспроизводятся в окне браузера – видимо, их открытие с помощью *MythTV* не предусмотрено.

Вариант «видеосёрфинга» интереснее (если его удастся настроить): видеоконтент на вашу машину поступает через RSS-видеоленту. По умолчанию лент не установлено, и вы получаете совершенно пустой экран; а чтобы добавить ленту, нужно продраться через загадочную систему меню. Но результат стоит трудов. Например, если вы подпи-

шетесь на шоу *Ted Talks*, *MythNetvision* будет автоматически подгружать новые выпуски. Добавив необходимые сервисы, можно «прочёсывать» их стандартной функцией поиска. Среди технических новинок – новый менеджер событий. Обычному пользователю он мало что даёт, но любители автоматизации процессов *MythTV* скриптами будут довольны. Теперь скрипт может срабатывать по запуску, остановке, окончанию, удалению записи. Регистрируются и другие события, происходящие при воспроизведении, а также при подключении новых компьютеров к сети. Новые привязки к Python придают *MythTV* небывалую гибкость. **LXF**

LINUX FORMAT Вердикт

MythTV 0.23

Разработчик: Айзек Ричардс

[Isaac Richards]

Сайт: www.mythtv.org

Цена: Бесплатно на условиях GPL

Функциональность	9/10
Производительность	8/10
Простота использования	4/10
Документация	6/10

» Солидное обновление. Мы бы рады оценить его выше, но пусть сперва разработчики покончат с ужасами настройки!

Рейтинг **7/10**

UndoDB 3.0

Не мечталось ли вам иной раз вернуть время вспять, для отладки зависшей программы? **Ник Вейч** нашел инструмент для таких путешествий.

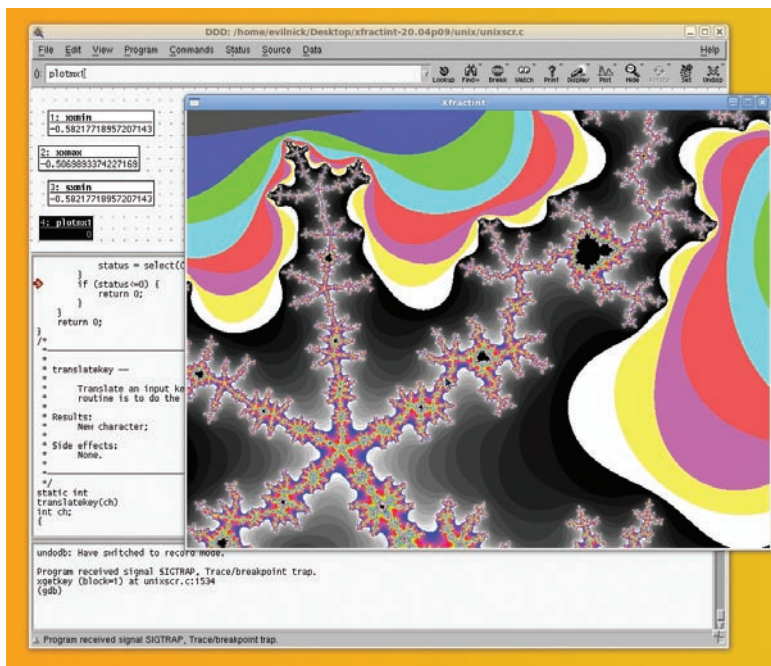
Вкратце

» Отладчик типа *GDB* — но с некоторыми радикальными отличиями.

Недосток обычного отладчика в том, что ошибка находится слишком поздно: программа уже «упала», нужно возвращаться и начинать сначала. Отладчик *UndoDB* обеспечивает запись каждого шага программы, что позволяет двигаться по контрольным точкам не только вперёд, но и назад.

Вы в полной мере оцените его мощь, осознав, что можно задать (новинка данной версии) обратные точки отслеживания [watchpoint]. Хм, на какой там стадии эта переменная вышла из-под контроля? Теперь можно без труда перемещаться назад во времени, сводя сложные трудоёмкие поиски к простой команде.

Одна из особенностей работы отладчиков, особенно с интерактивными графическими программами — перегрузка процессора. Ведь записывается всё и вся, и приложение «тормозит» иногда до пол-



» *UndoDB* замедляет просмотрщик фракталов *Xfractint* всего на 50%, что весьма неплохо — причём для учета замедления имитирует штатные часы.

«Среди новинок — способность отладки 64-битных приложений.»

ной остановки. Прежде *UndoDB* с трудом справлялся с отладкой сложных графических или ресурсоемких программ, но эта версия сильно улучшена. Ещё одна беда — при записи большого количества информации переполняется память. Теперь запись идёт в кольцевой буфер: при его

переполнении наиболее старые данные удаляются.

Только попробуйте

Как правило, программисты неохотно меняют инструменты. Поэтому в *UndoDB* хитроумно используется стандартный набор команд *GDB*, так что подмены и не заметишь (особые команды применяются лишь с многопоточными приложениями).

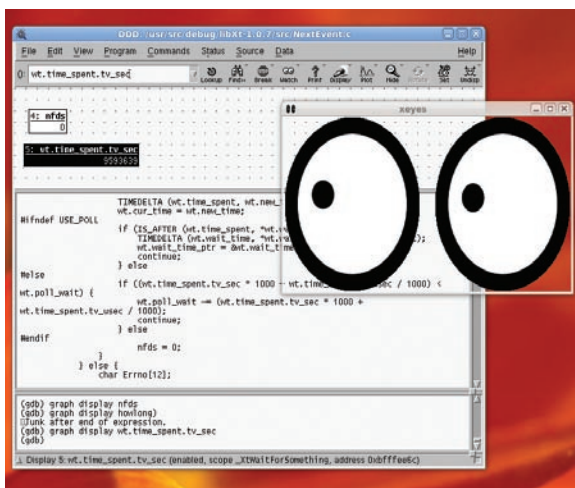
UndoDB работает также со средством просмотра отладочных данных *DDD* — достаточно лишь ввести переключатель `--debugger undodb-gdb`, и всё должно быть замечательно. Только в консоли придётся вводить команды именно *UndoDB*. Если вы проводите отладку в другой среде, попробуйте экспериментальную поддержку *GDB 7.0*: с этим дополнением отладчик встраивается в пакеты типа *Eclipse*.

Другие новинки включают способность отлаживать 64-битные приложения, или при работе на 64-битных системах отлаживать 32-битные программы: отпадает необходимость настраивать несколько экземпляров среды программирования.

Новая версия отладчика поддерживает команды процессоров SSE4. Их применение не ограничивается мульти-

медиа-приложениями и включает функции шифрования и оптимизации 64-битных систем.

Возможности, которыми *UndoDB* может обогатить вашу среду программирования, бесценны (сколько стоит день просрочки проекта, вы знаете), но разработчики открытого ПО получают стартовую версию бесплатно. Остальным необходимо приобрести лицензию, но это ничтожная плата за великолепный инструмент. **LXF**



» Кроме интерфейса командной строки, *UndoDB* работает с графическими инструментами отладки, например *DDD*.

LINUX Вердикт
FORMAT

UndoDB 3.0
 Разработчик: Undo Software
 Сайт: http://undo-software.com/undodb_v3.html
 Цена: Для некоммерческого использования — бесплатно. Стартовая версия на 1 год — \$95

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

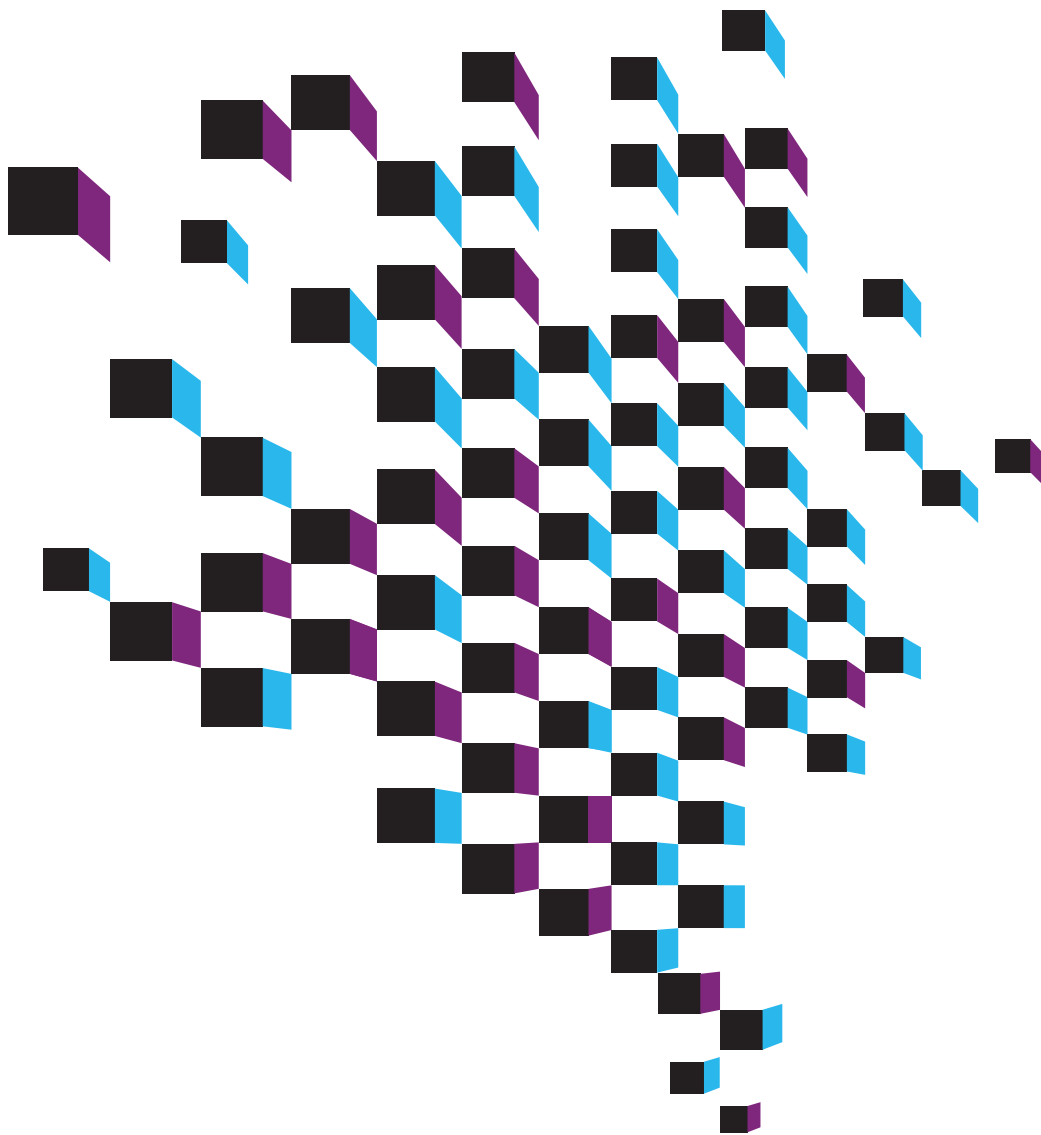
» По ценности для программиста эта вещь стоит второй после машины времени.

Рейтинг 9/10

Нестандартные решения зависят от нестандартных задач

Растущие потребности современных серверов в энергии не сводятся только к росту затрат. Все чаще они прямо влияют на повседневную работу компании. Согласно недавнему исследованию, около 50% организаций сталкивались с простоями в работе ИТ-систем, вызванными проблемами с питанием и охлаждением серверов¹. Особенности архитектуры IBM BladeCenter® HS22 позволяют повысить эффективность работы на всех уровнях. Это и высокоэффективная конфигурация, и процессор Intel® Xeon® серии 5500, и передовое программное обеспечение, такое как IBM Systems Director, динамично отслеживающее энергопотребление, а также встроенные датчики, позволяющие оптимизировать охлаждение. Благодаря всему этому экономия энергии может достигнуть 93% по сравнению с предыдущими поколениями стоечных серверов. Хотите узнать, как окупить инвестиции всего за три месяца?² Посетите ibm.com/hs22/ru

Системы, программное обеспечение и сервисы для улучшения экологии планеты.



Реклама



¹ Прогноз расходов на энергопотребление серверов в мире на период с 2008 по 2012 гг. – Анализ рынка № 215870, составленный IDC, том 1, декабрь 2008 г.

² Данные по окупаемости вложенных средств и экономии энергии основаны на расчетах, выполненных, исходя из сценария с коэффициентом консолидации 11:1 для 166 сокетных серверов Intel высотой 1U по отношению к 14 серверам BladeCenter HS22, с учетом экономии на расходах на энергию, лицензии на программное обеспечение и другие текущие расходы. Расходы и объем сэкономленных средств зависят от конкретной конфигурации и среды. Более подробная информация приведена на www.ibm.com/smarterplanet/claims. IBM, логотип IBM, ibm.com, BladeCenter и Systems Director VMControl 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. © 2010 IBM Corporation. Все права защищены.

Сравнение

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

Web-браузеры

По Нику Вейчу, это самый важный инструмент, но какой именно взять себе?



Про наш тест...

Все браузеры тестировались на компьютере с 3-ГГц процессором Core 2 Duo и 4 ГБ ОЗУ, под Fedora 12.

Мы провели ряд доступных и популярных тестов и усреднили результаты по каждой позиции. Следует отметить, что многие из этих тестов часто используются для оптимизации браузеров (как, например, SunSpider), поэтому мы включили самые разные. Не стоит цепляться за каждую миллисекунду — гораздо важнее ощущения пользователей и общая скорость.

По возможности, все тесты скорости проводились на локальных копиях web-страниц. В некоторых испытаниях это не являлось строго обязательным критерием, поскольку хронометраж начинался только после загрузки страницы. Если вы собираетесь проделывать тесты сами, важно выполнить их несколько раз.

Наша подборка

Epiphany	c.16
Firefox	c.15
Flock	c.15
Google Chrome	c.18
Konqueror	c.17
Midori	c.16
Opera	c.17
SeaMonkey	c.18

Web-браузер становится важнейшим компонентом программного обеспечения для настольных ПК, если еще не стал. Интернет — не только богатый источник информации, но и проводник в целый мир сетевых приложений и взаимосвязанного облака услуг, охватывающих широкий диапазон нового опыта работы с ПК. Когда вы делаете покупки, вам нужна безопасность; когда вы работаете, вам нужна надежность; а когда развлекаетесь, вам нужна скорость и совместимость с различными типами медиа. Помимо этого, большинство сайтов работают с использованием технологий JavaScript и Ajax, что создает

дополнительные проблемы для обработки информации в браузере. Создание браузера, умеющего справляться со всем этим и одновременно простого в использовании — довольно трудная задача.

Наше правило выбора только стабильного выпущенного ПО пришлось отставить, поскольку многие браузеры находятся в хронической стадии бета, постоянно улучшая время загрузки, JavaScript и все такое. Таким образом, мы

использовали последние опубликованные версии. В основном это касается *Chrome*, *Opera* и *Firefox*.

Принцип отбора, какое именно ПО включить в наше Сравнение, весьма прост: это наиболее популярные браузеры Linux, активно разрабатываемые и используемые. Мы включили только отлаженные версии браузеров, доступные уже сейчас — не пре-альфа и не «ночные» сборки.

«Web — окно в мир сетевых приложений и взаимосвязанных облачных сервисов.»

Firefox

Когда-то — дитя новой революции в Web, но не прошли ли его времена?

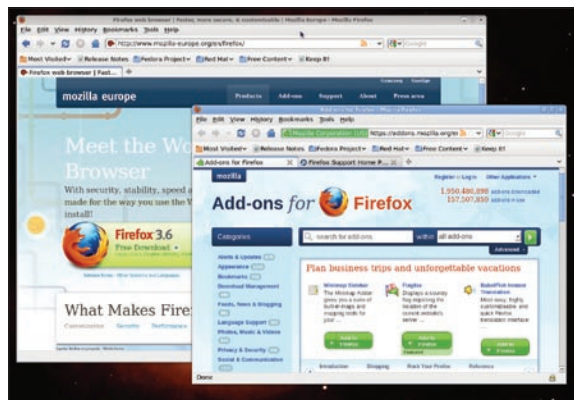
Существует описание *Firefox* как блестящего спортивного автомобиля со всякими эзотерическими аппаратными наворотами, приваренными снаружи. В качестве аналогии это неплохо. Первоначальным импульсом к разработке *Firefox* (см. LXF66) было создание отлаженного, быстрого и эффективного браузера, который не усложнял бы интерфейс пользователя и не страдал излишками кода, используемого лишь меньшинством и снижающего скорость работы. Естественно, в итоге браузер сам вскоре обзавелся квадрильоном расширений.

Стремительный взлет *Firefox* (ему удалось получить около 20% доли рынка браузеров за первый год, и сейчас считается, что это клиент, потребляющий почти половину всего web-трафика в мире) показывает, что гладкий и четкий стиль был хорошим началом. *Firefox* завоевал популярность благодаря не только скорости работы, но и инновациям, и он стремился к реальному соблюдению стандартов в мире, где браузеры вроде *Internet Explorer*

хотели заставить Web работать по своему сценарию.

Но это было в прошлом; а что же *Firefox* сделал для нас за последнее время?

Большинство из последних изменений выглядят косметическими, но есть также и технические новшества: например, поддержка Web Open Font Format, недавней разработки, упрощающей встраивание загружаемых шрифтов таким образом, что их разработчики сыты, а трафик цел. *Firefox* также внедряет довольно много новых технологий HTML 5, с поддержкой OGG-контейнеров и формата WebM от Google, MathML и прочее. *Firefox* отнюдь не почил на своих честно заслуженных лаврах. В сочетании с отличной политикой безопасности и поразительным количеством потенциальных настроек, *Firefox* — хороший выбор для web-обозревателя.



» Дополнения *Firefox* включают множество полезных вещей и настроек.

«Firefox отнюдь не почил на своих честно заслуженных лаврах.»

LINUX FORMAT **Вердикт**

Firefox

Версия: 3.6.3
Сайт: www.getfirefox.org

» Устойчивый и надежный, гибко настраиваемый и на диво проворный.

Рейтинг 9/10

Flock

Так называемый социальный браузер не лишен сюрпризов.

Flock появился примерно тогда же, когда и *Firefox*, но руководствовался другим. *Flock* концентрируется на том, для чего многим людям только и нужен Интернет: Facebook, YouTube, Flickr, MySpace и различных службах электронной почты. Идея заключалась в том, чтобы построить ряд инструментов, необходимых для работы — редактор блогов, загрузчик фотографий — сохранив возможность потреблять это средство на «нормальных» web-сайтах. С виду он похож прямо таки на центр управления мультимедиа, что не слишком далеко от истины.

К сожалению, боковая панель, предназначенная для слежения за Twitter или Flickr, становится не слишком отзывчивой, создавая дополнительную нагрузку на процессор (и канал) ради получения свежей информации о последних твитах или изображениях. Конечно, нужно мириться с тем фактом, что для показа в браузере и web-страниц, и записей Twitter информация должна загружаться и обрабатываться. Наверное, приоритеты можно было бы

скорректировать получше, но для пользователя соединения в 1 МБ или медленнее это плохой компромисс.

С точки зрения производительности и рендеринга, *Flock* частенько плетется в хвосте. Хотя он все еще в стадии разработки и использует ту же кодовую базу, что и Mozilla *Firefox*, год затишья оставил его стоять на том же месте.

Flock вполне подойдет в качестве браузера, если вы днями напролет выясняете, что новенького происходит на Facebook, но большую часть функций *Flock* можно вывести в расширения (хотя столь же мучительно медленные) или, возможно, в специализированные приложения. У него есть интересные и даже инновационные интерфейсные решения, но они уже изрядно устарели.



» Контроль нескольких социальных сервисов требует некоторых затрат.

«С виду Flock похож на центр управления мультимедиа.»

LINUX FORMAT **Вердикт**

Flock

Версия: 2.5
Сайт: www.flock.com

» Реально иначе подходит к работе с web, но позабыт-позаброшен.

Рейтинг 5/10

Midori

Использует Webkit, черпает идеи отовсюду, а назван в честь напитка.

Название браузера в честь ликера – неперменного компонента праздничной выпивки – может показаться странноватым, но, возможно, непреднамеренная ссылка на коктейль выдержит критику. *Midori* – браузер, рассчитанный в первую очередь на легковесность, но обладает также и массой новейших свойств. Так вот, в его рецепт включены некоторые фирменные функции других браузеров – Speed Dial от *Opera*, неперменная панель поиска *Google* и расширения, как у *Firefox* (хотя и в значительно меньшем количестве).

Может ли смесь заимствованных идей породить хороший браузер? Ну, по крайней мере легковесность тут обеспечили. Хотя потребление памяти и не совсем минимальное, но это не злостный пожиратель процессора или трафика. Web-страницы отображаются при помощи *WebKit*, и скорость у браузера такая же, как и у других обозревателей на основе этого движка.

Имеется выбор расширений для браузера, допускаются доделки на свой вкус,

а также поддерживаются модули расширения в стиле *Netscape* для воспроизведения различных медиа-файлов через *Totem* и *Rhythmbox*. Любопытно, что в панели URL внедрена функция «введите что-нибудь, и если это не URL, то найдите его», но есть и отдельный гаджет web-поиска. Очень удобна функция корзины – этот значок на главной панели инструментов покажет вам вкладки, которые вы недавно закрыли, и позволит заново открыть их, выбрав из списка. Многие браузеры позволяют заново открыть URL из предыстории, но мало где это столь же легко.

Хотя *Midori* вполне прилично выполняет все функции, веских причин предпочесть его *Epiphany* (браузеру по умолчанию в *Gnome*) – и уж конечно, «старшим пациентам» – нет.

«Рассчитан на легковесность, но обладает и новейшими свойствами.»



➤ Проворный браузер, соответствующий стандартам благодаря *WebKit*.

LINUX Вердикт
FORMAT

Midori

Версия: 0.2.2
Сайт: www.twotoasts.de

» Среднячок, но берет своей нетребовательностью к ресурсам.

Рейтинг 5/10

Epiphany

Этот почти анонимный браузер *Gnome* не стоит обходить стороной.

У всех пользователей рабочего стола *Gnome*, вероятно, уже установлен *Epiphany*, а они и не заметили. Будучи браузером по умолчанию, он обычно упоминается в меню просто как «браузер», и даже вызов «О программе» даст вам мало сведений о его происхождении. *Epiphany* раньше базировался на движке *Gecko*, но одно из преимуществ ПО с открытым исходным кодом – то, что вы можете менять компоненты, как захотите. Браузер *Midori* и появился как альтернатива *Epiphany* для пуристов *Gnome*, потому что внедрил *WebKit*. *Epiphany* малость засиделся на старте, но сейчас также применяет *WebKit*, и эти браузеры не особо отличаются по скорости и совместимости.

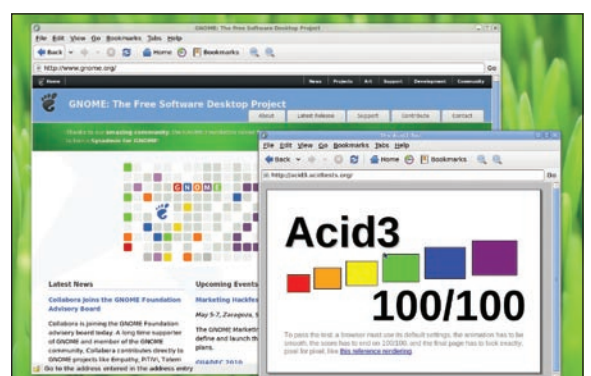
Epiphany ратует за концепцию «меньше значит больше» для настольного программного обеспечения, и, следовательно, в нем нет фолиантов страниц с опциями настройки или параметрами, задаваемыми пользователем. Это упрощает работу с ним, но и слегка расстраивает тех,

кто, например, с удовольствием указал бы предпочтения для всплывающих окон в зависимости от сайтов.

Мощь *WebKit* сквозит везде, и хотя *Epiphany* в основном привлекает людей, для которых настройки – не главное (в отличие от некоторых других браузеров в данном Сравнении), он превосходно справляется с тестами и отнюдь не производит впечатления медлительного при просмотре страниц.

Хотя иногда *Epiphany* вроде бы обгоняет *Midori* в некоторых тестах, с точки зрения скорости эти два браузера работают практически одинаково (чего и следовало ожидать, так как и внутри они почти идентичны), и их результаты совпадают с точностью до погрешности измерений.

«В Gnome уже установлен Epiphany, а пользователи и не заметили.»



➤ Постарайтесь не переоценивать его возможности, поскольку это браузер по умолчанию.

LINUX Вердикт
FORMAT

Epiphany

Версия: 2.28.2
Сайт: <http://projects.gnome.org/epiphany>

» Только потому, что он прост, не стоит считать его бесполезным.

Рейтинг 5/10

Konqueror

Единственный практик web с правом чтения-записи, *Konqueror* на страже.

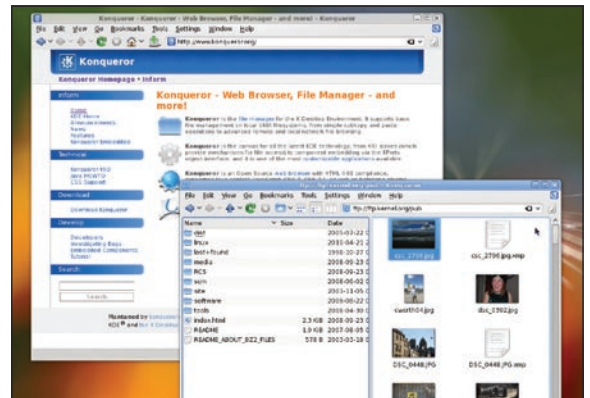
Долгое время *Konqueror* был, пожалуй, лучшим приложением в KDE – не только как способный и удовлетворяющий стандартам браузер (хотя на том этапе никто не понял, как поддерживать все те проприетарные расширения), но также и прекрасный файловый менеджер. Эти две функции сидели бок о бок, а внедрение компонентов KParts означает, что *Konqueror* рассматривает практически все просто как объекты рендеринга и взаимодействия, будь то локальный каталог, удаленный каталог FTP, разделяемый ресурс *Samba*, сайт или что угодно. Времена меняются, и хотя старый *Konquy* все еще браузер по умолчанию для KDE, его файловые функции перехватил *Dolphin*. Его движок – совсем другая история: им воспользовались в Apple для создания *WebKit*, оснащения для *Safari* и многих других браузеров. *Konqueror* остался едва ли не единственным браузером, который по-прежнему придерживается визуализации *KHTML*, но почему бы и нет? Он, может, и не самый быстрый инструмент, но до сих пор проделывает

серьезную работу по части соответствия стандартам и поддерживает большое количество HTML 5.

Голая скорость загрузки и рендеринга страниц – это одно, а скорость работы с тем, что вам нужно – совсем другое. КЮ *Konqueror* и умение работать как файловый менеджер делают его гораздо эффективнее при выгрузке файлов на FTP или разделяемый ресурс WebDav, благодаря отлаженному взаимодействию. Большинство браузеров создаются как потребители Интернета, а *Konqueror* считает web еще одним ресурсом, для чтения или записи информации.

На фоне KDE, особенно по сравнению с *Firefox*, *Chrome* и Co, *Konqueror* также весьма нетребователен к ресурсам, потому что многие необходимые ему библиотеки уже загружены.

«Долгое время Konqueror был, пожалуй, лучшим приложением в KDE.»



» *Konqueror* – ценный инструмент для работы и в локальной сети, и в web.

LINUX FORMAT **Вердикт**

Konqueror
 Версия: 4.4.2
 Сайт: www.konqueror.org

» Малость тормозит, но зато весьма разносторонний.

Рейтинг 6/10

Opera

Opera выделяется необычным подходом к составу браузера.

Будучи приложением с закрытым исходным кодом (хотя теперь и бесплатным), *Opera* имеет собственный движок рендеринга (*Presto*) и JavaScript, из-за чего результаты его производительности в проведенных нами тестах варьируются. *Opera* также придерживается своего личного мнения об опыте пользователя и работе Интернета. Например, *Opera Turbo* – остроумная технология сжатия, повышающая скорость загрузки многих сайтов (хотя это и невозможно проверить эмпирически, но на некоторых сайтах, видимо, все-таки работает).

Opera Unite – еще одна интересная особенность: создается своего рода персональная сеть между пользователем и его друзьями (тоже пользователями *Opera*), для обобщения файлов, ссылок и другой информации. Короче, разработчики много думают о работе в браузере с точки зрения пользователя.

Это очень способный браузер с необходимой безопасностью, персонализацией и конфиденциальностью. Для работы

с модулями расширения он привлекает библиотеки в стиле *Netscape* и предлагает скорее виджеты, чем расширения. В чем разница? Виджеты в меньшей степени меняют работу браузера: это узко направленные инструменты или клиенты web-сервисов типа информации о погоде.

Несмотря на уверения разработчиков, по результатам наших тестов *Opera* нельзя назвать самым шустрым из браузеров. Но он выглядит очень быстрым и отзывчивым. Испытания JavaScript доказали эффективность его движков, но по жизни вы вряд ли посетите много сайтов, столь интенсивно использующих JavaScript, чтобы отличие стало заметным.

Версия 10.50 для Linux собирается выйти уже давно, но для тестирования вы можете скачать бета-релиз хоть сейчас.

«Будучи закрытой, Opera имеет собственный движок рендеринга.»



» В *Opera* имеются полезные инструменты – например, средство отладки JavaScript.

LINUX FORMAT **Вердикт**

Opera
 Версия: 10.10
 Сайт: www.opera.com

» Отлично работает с Интернет, но заявления насчет скорости слабо обоснованы.

Рейтинг 6/10

Google Chrome

Не без недостатков, но не близок ли он к совершенству?

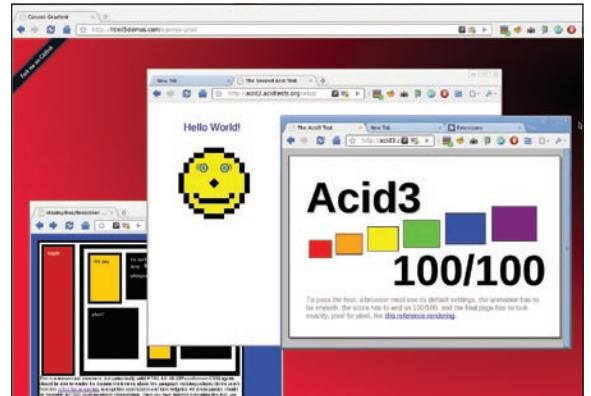
Проясим ситуацию: тому, что *Chrome* работает быстро, дивиться нечего. На это были брошены почти все усилия с момента первого запуска браузера, от бесконечных ухищрений с движком JavaScript до добавления новых технологий, типа предзагрузки доменных имен (DNS). Иногда эти преимущества проявляются не во всех аспектах: хороший пример – та же предзагрузка DNS. В этом случае, браузер считывает все ссылки на странице и иницирует процесс поиска запроса DNS. Вы еще только собрались нажать на ссылку, а результат уже в кэше. Сокращение времени поиска – это хорошо, но заодно попусту расходуется интернет-трафик. Вообще-то большинство пользователей это не волнует: они, скорее всего, и не подозревают, что происходит.

Любой, кто открывал сразу 250 вкладок и на последней уронил *Firefox*, поймет, как неприятно восстанавливать все заново. Именно поэтому *Chrome* порождает свой процесс для каждой вкладки – и если что-то пойдет не так, вы не потеряете

все. Это система, которая хорошо работает и позволяет избежать катастрофы, когда что-то выходит из-под контроля.

Chrome – не лучший браузер всех времен и народов. Случается, что вкладки падают без видимых причин. Есть вопросы к простоте использования. Модулей расширения не так много, как в *Firefox*, и по ресурсам *Chrome* бесспорно расточителен. Зато это быстрый, безопасный и удобный инструмент просмотра. И, говоря «быстро», мы имеем в виду – очень быстро. Возможно, он был оптимизирован для прохождения тестов, но и при обычном использовании тоже очень и очень быстр. Минималистский интерфейс максимизирует охват просмотра web, и хотя к нему надо немного привыкнуть – меню явно «суетливее» обычного – это хороший компромисс для многих пользователей.

«Это быстрый, безопасный и удобный инструмент просмотра.»



➤ Самый быстрый браузер под Linux со множеством полезных функций.

LINUX FORMAT **Вердикт**

Google Chrome

Версия: 5.0.375.55 beta
Сайт: www.google.com/chrome

» Он не совершенен, но движется в этом направлении.

Рейтинг 10/10

SeaMonkey

Его идея не из тех, что просто лежат себе догнивая.

Изначально браузер *Netscape* был «интернет-комплексом», объединявшим функции web-браузера, почтового клиента и редактора HTML. *SeaMonkey* развивает эту идею, добавляя IRC-чат, чтение новостей, RSS-агрегатор и добавочные инструменты разработки. Поскольку он построен в основном на коде Mozilla, то эксплуатирует те же преимущества технологии и производительности.

Как и *Flock*, теоретически он совместим с расширениями *Firefox*, но с теми же ограничениями – многие расширения ориентированы на конкретные функции пользовательского интерфейса *Firefox* или вещи, которых в *SeaMonkey* попросту нет.

К результатам *SeaMonkey* по эффективности следует относиться с оговоркой: пока мы это пишем, готовится выпуск версии 2.1, и в него, вероятно, войдет более современный код *Mozilla*, что повлияет на увеличение скорости и на совместимость.

Пользовательский интерфейс немного коряв, но очень удобен в плане функцио-

нальности. Отчасти это шаг назад с точки зрения дизайна, но многие люди полюбили доходчивые большие иконки и простоту обработки вкладок, так что это не стоит считать недостатком.

Стратегия *SeaMonkey* выглядит довольно-таки здоровой. Для многих людей при работе с Интернетом большую роль имеет функциональность, а не развлекательные аспекты. Склейка всех нужных инструментов в один пакет – идея в самом деле неплохая. Компонент электронной почты хорош, и хотя страница создания сообщения может не хватать некоторых полезных инструментов автономного редактора, он прост в использовании и работает достойно. Для web-разработчиков предусмотрены консоль JavaScript и отладчик.

«Склейка всех инструментов в один пакет – идея неплохая.»



➤ Простой в использовании интегральный пакет имеет свои преимущества.

LINUX FORMAT **Вердикт**

SeaMonkey

Версия: 2.0.4
Сайт: www.seamonkey-project.org

» Прекрасный универсал; выглядит старовато, но работает прекрасно.

Рейтинг 8/10

Web-браузеры

Вердикт

Chrome 10/10

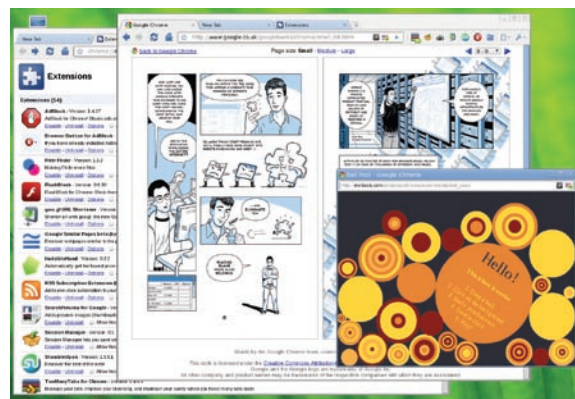
Как и в большинстве категорий программного обеспечения, не существует и браузера, подходящего абсолютно всем. Некоторым не хватает гибкости *Konqueror* и его превосходной подсистемы КЮ. Одни гонятся за простотой, как *Midori* и *Eipiphany*, а другие предпочитают «все включено», как *SeaMonkey*.

Opera работала под Linux, начиная с версии 4.0, выпущенной в 2000 году. Это, пожалуй, самый нетипичный браузер Сравнения – просто потому, что разработка его закрытая. Был момент, когда *Opera* предлагала лучшее, но Linux-версии немного отстают, что затрудняет создание конкурентных преимуществ.

Firefox – действительно отличный браузер, и до сих пор это наиболее популярный выбор среди пользователей Linux. Если вы счастливы с *Firefox* и потеря любимых модулей расширения будет для вас трагедией, нет большой нужды что-то менять.

Победителем нашего теста стал *Chrome*. Он побил все прочие браузеры не только по скорости, но и по совместимости. Восхищаясь скоростью *Chrome*, не забудем и о прекрасном ассортименте встроенных инструментов разработчика. Обычных пользователей они, возможно, не привлекут, но для тех, кто создает сложные сайты, графики хронометража и инструменты профилирования, будут реальным подспорьем. Интересно, ассимилируют ли эти технологии другие проекты, коль скоро *Chrome* – открытая разработка.

Сейчас заметно, что битва браузеров, вероятно, стала намного острее, чем



» Встроенные инструменты разработчика в *Chrome* – мечта хакеров.

в последние годы. С переходом к HTML 5 и взрывным выходом на сцену *Chrome* с его агрессивным ускорением, темп инноваций и изменений в технологиях браузеров в ближайшее время не замедлится. **LXF**

Обратная связь

Есть много причин, по которым можно предпочесть один браузер другому. Мы хотели бы узнать, как выбрали вы. Что лучше: маленький и легкий или большой и сложный? Сообщите нам по адресу: letters@linuxformat.ru

«Битва браузеров стала намного острее, чем в последние годы.»

Сравнительная таблица характеристик

Браузер	Chrome	Eipiphany	Firefox	Flock	Konqueror	Midori	Opera	SeaMonkey
Движок	WebKit	WebKit	Gecko	Gecko	KHTML	WebKit	Presto	Gecko
Соответствие Acid3	100%	100%	94%	72%	88%	100%	99%	93%
Время выполнения Acid3 [мс] ¹	430	2030	2020	4270	5900	2290	8250	540
Тест JavaScript celtickane ²	424	118	147	88	67	119	62	126
Тест SunSpider ³	453	2073	1252	3544	4443	2109	5708	1576
Тест Dromaeo ⁴	5817	105	85	33	42	106	50	68
Тест V8 ⁵	4570	530	333	170	Провален	529	152	247
Тест jQuery/SlickSpeed ⁶	29	50	90	195	159	53	90	110
Медиа-тест, потребление CPU ⁷	5%	13%	5%	Провален	13%	14%	5%	5%
Тест видео HTML 5 ⁸	✓	✓	✓	✗	✗	✓	✗	✗
Потребление памяти [байт] ⁹	188585	58780	113560	241782	106988	153520	127654	198954

¹Время выполнения теста Acid3. Чем меньше, тем лучше. ²Чем больше, тем лучше. Усреднено по 10 тестам. <http://jsbenchmark.celtickane.com/Run.aspx> ³Чем меньше, тем лучше. www2.webkit.org/perf/sunspider-0.9/sunspider.html ⁴Чем больше, тем лучше. Усреднено по 4 тестам (тест очень долгий!) <http://dromaeo.com/?dromaeo> ⁵Тесты V8 созданы и поддерживаются командой Google Chrome. <http://v8.googlecode.com/svn/data/benchmarks/v5/run.html> ⁶<http://mootools.net/slickspeed> ⁷Воспроизведение видео высокой четкости – потребляемая доля процессора. Чем меньше, тем лучше. ⁸<http://html5demos.com/two-videos> ⁹Тест по 10 сайтам на жестком диске.

Linux vs iPhone

Грэм Моррисон изучил три смартфона на базе Linux, обыгравшие Apple на их же поле.

Корпорация Apple, несомненно, проделала огромную работу, создавая свой iPhone — и в плане ПО, и в плане оборудования. Она одной левой превратила некогда раздутый и застойный рынок смартфонов в очень прибыльное дело. Всем изготовителям пришлось заняться переосмыслением своей конкурентной стратегии. ОС Linux тоже получила возможность проявить себя.

Поразительно — Linux удалось добиться впечатляющих успехов на рынке мобильных телефонов. Ему удалось стать эквивалентом ОС для процессоров ARM (LXF132), завоевавшей доминирующие позиции на рынке благодаря своей фантастической многофункциональности, производительности и дешевизне.

Этот успех свидетельствует в пользу ОС Linux и методов ее работы. Он доказывает, что на аппаратном уровне соревнование стилей и брендов на рынке мобильных телефонов значит очень мало. То, чего хотят добиться изготовители, дает им опера-

ционная система, и именно поэтому производители мобильных устройств выбирают Linux.

Мы протестировали три смартфона с операционной системой Linux, которые по своим ценам дают iPhone

большую фору. Это Palm Pre, использующий WebOS, основанный на Maemo 5 N900 фирмы Nokia и HTC Legend, работающий под управлением Android. Обойдется любой из этих аппаратов дешевле iPhone, а сегодня они побеждают его и по многим другим важным показателям. Какой же из них вы предпочтете?

«Linux удалось стать эквивалентом ОС для процессоров ARM.»



Аппаратная начинка

Palm Pre

Заняв нишу «персональных цифровых помощников [PDA]», фирма Palm несколько лет пыталась переключиться на смартфоны, внедряя свои старые наработки в мобильный телефон типа BlackBerry. С выходом аппарата Palm Pre, компания наконец-то добилась цели. Плавные контуры и гляцевый черный дизайн устройства существенно повлияли на рынок, и к тому же оно имело приличный размер и приятный «экстерьер».

Добавление вертикально прокручиваемой клавиатуры, которая выдвигается снизу устройства мягким щелчком – шаг вперед по сравнению со старыми версиями устройств от Palm. Однако клавиатура все равно не слишком удобна, если у вас крупные руки и толстые пальцы, особенно учитывая то, что она изначально сконструирована для работы двумя пальцами. Заставить большой палец правой руки нажать кнопку H после того, как большой палец левой руки только что нажал кнопку t, довольно проблематично, но к этому привыкаешь, как у всех подобных устройств.

«Экран Nokia N900 обставит и Palm Pre, и iPhone и по разрешению, и по четкости.»

Нижняя часть дисплея используется для вывода обычных значков, а также наиважнейшей кнопки Вызов [Call]. Нажав на нее, можно либо позвонить кому-то из адресной книги, либо ввести номер вручную. В целом, устройство выглядит более удобным и управляемым по сравнению со старыми аппаратами фирмы HTC или Nokia N900.

Оригинальный Palm Pre также снабжается отдельной кнопкой между емкостным сенсорным экраном (с диагональю 3,1 дюйма, разрешением 320×480 и поддержкой технологии Multitouch) и выдвигной клавиатурой. Нажатие на нее приводит к мгновенному свертыванию всех выполняющихся приложений и дает возможность запустить что-либо еще. Эта функция, безусловно, дает выигрыш по сравнению с однозадачным iPhone. В Palm Pre Plus данная кнопка удалена, но зато эта новая версия добавила 512 МБ внешней памяти и удвоила дисковое пространство – до 16 Гб флэш-памяти. Что еще важнее, в новой версии устройства чуть увеличено расстояние между кнопками клавиатуры, упрощая работу людям с большими руками. К сожалению, у нас не было возможности протестировать устройство на практике и проверить, стало ли оно действительно удобнее.

Аккумуляторы можно менять!

Внутри корпуса – процессор Omар 3430 ARM и аккумуляторная батарея, достаточная для непрерывной работы в течение пяти часов интенсивных разговоров. Сообщается, что заряда батареи будет достаточно в течение тринадцати дней в режиме ожидания звонка. В отличие от наглухо запечатанного iPhone, в Palm Pre вы при необходимости можете заменить батарею. Имеется также сопроцессор C64x – он ускоряет воспроизведение файлов OGG Theora и поддерживает новый видеоформат VP8 от Google.



Поддержка графики обеспечивается интегрированным ядром Power VR SGX 530, чуть менее мощным, чем чип PowerVR, вмонтированный в iPhone 3GS (за достоверность сравнения поручиться нельзя, потому что Apple не публикует точных спецификаций на свои устройства). Наконец, современный телефон не полон без всевозможных сенсоров. Palm Pre оснащается акселерометром, датчиком близости (полезен при звонках) и чипом GPS-навигации. Имеется неплохая встроенная камера с матрицей 3,2 Мпикс, светодиодной вспышкой и геотэггингом, но, к сожалению, без функции автофокуса. Версия ОС 1.4, выпущенная в феврале, добавила съемку и просмотр видео.

► При подключении к компьютеру Palm Pre виден как накопитель, и с ним удобно обмениваться музыкой и фотографиями.

Nokia N900

Самое яркое впечатление от этого аппарата – его экран. Он легко обставит и Palm Pre, и iPhone – и по разрешению, и по четкости. Мы считаем, что разрешение 800×840 – большой шаг вперед, и будущим версиям Palm Pre и iPhone о таком только мечтать. Но у него есть огромный минус: он резистивный, а значит, не поддерживает технологии Multitouch. Nokia, конечно, хорохорится, но они явно чувствуют, что это серьезный недостаток. Причем он еще и бросается в глаза из-за встроенного стилуса, как у старых КПК с ОС Windows Mobile, хотя можно поспорить, что управление пальцами удобнее.

Полный вперед

Процессор у N900 тот же, что и у устройства от Palm – Omар 3430 с сопроцессором C64x. Однако здесь он работает на полной тактовой частоте – 600 МГц, а не 500, как в Palm Pre. Память собрана отчасти с бору по сосенке: физической – 256 МБ, но еще до 768 МБ внешней флэш-памяти можно употребить на файл подкачки для процессов, запущенных в фоновом режиме. Как и в КПК, работающих под Windows Mobile, здесь имеется слот Micro SDHC, и устройство может адресовать до 32 Гб памяти.

Батарея N900 несколько мощнее, чем в устройствах от Palm (1320 мА·ч по сравнению с 1150 у Palm Pre), но повышенная так-

»



► Встроенная камера Nokia N900 имеет матрицу 5 Мпикс и линзу от Carl Zeiss.



товая частота берет свое. Нам удалось протянуть на N900 около 8 часов работы в нормальном режиме; у Palm Pre рабочего времени больше чем вдвое.

Устройство, как полагается, снабжено осевыми сенсорами для переключения экрана между ориентациями портрет–ландшафт, а также датчиком близости и приемником GPS, аналогично Palm Pre и большинству устройств с Android. Еще одно достоинство – возможность подключения аппарата к телевизору через прилагающиеся аудио- и видеоканалы.

Но важная техническая характеристика N900 – выдвигающаяся Qwerty-клавиатура. Она оснащена задней подсветкой и просторнее клавиатуры Palm Pre, однако по-прежнему тесна, и, хуже того, ее кнопки организованы в виде матрицы. Например, кнопка Q находится прямо над кнопкой A, а кнопка P – в собственной колонке далеко вправо. У Palm Pre ряды клавиш смещены, и ими удобнее пользоваться.

Небольшая фаска, добавленная на заднюю панель из-за камеры, означает, что N900 не укладывается на плоскую поверхность, затрудняя печать на клавиатуре; правда, фаска скрывает и подпорку для просмотра видео на телефоне. Добавьте к этому отсутствие функции исправления ввода, и вы невольно задумаетесь, являются ли дополнительный вес, размер и сложность физической клавиатуры наилучшим техническим решением для N900.

HTC Legend

Существуют десятки устройств на платформе Android, с различными спецификациями, от объема памяти до экранного разрешения. Выбор конкретной модели обычно диктуют цена и производительность, но доминирует изготовитель HTC. Данная фирма спроектировала смартфон Google Nexus One – сейчас это флагманский продукт; но у нее есть и собственные устройства на базе Android. Новейшие из них – HTC Desire и HTC Legend.



► HTC Legend под управлением Linux – серьезное улучшение по сравнению с устройствами на базе Windows Mobile.

HTC Legend показывает, как далеко вперед ушли разработки фирмы HTC со времен выпуска iPhone. HTC Legend намного меньше, тоньше, светлее и имеет лучший, более яркий емкостной OLED-экран, хотя и с тем же разрешением. Самое поразительное – устройство выглядит как картинка из буклета Apple, потому что его корпус изготовлен из цельного куска алюминия.

Черные клавиши

На передней панели устройства – пять кнопок: одна для вызова «домашнего» экрана, вторая – для открытия меню, третья – для возврата к предыдущему пункту и еще одна – для поиска. Пятая кнопка, расположенная под ними, действует аналогично нижней части оптической мыши. Как и ожидалось, она нажимается, но можно и водить пальцем по ее поверхности, перемещая невидимый курсор по экрану.

Устройство оснащено камерой 5 Мпикс с автофокусом и вспышкой; предусмотрен разъем микро-USB для подключения к компьютеру. Ну, а внутри – стандартный набор датчиков: GPS, близости, силы тяжести и окружающей освещенности. Имеется потайной слот для карты памяти MicroSD; ее можно прикупить для хранения приложений, фотографий и просто данных.

В отличие от ранних версий, выдвигающаяся клавиатура нет. Экранная клавиатура Android похожа на виртуальную клавиатуру iPhone, хотя из-за небольшого размера экрана кнопки притиснуты еще ближе друг к другу. Тактильная обратная связь, эффект быстрой вибрации устройства при нажатии на виртуальную кнопку – полезное дополнение, помогающее при быстрой печати.

► Аппаратная клавиатура – это хорошо, но данный вариант использовать сложно.



Сравнительные характеристики

Модель	ОС	Экранное разрешение	Вес, г	ОЗУ, МБ	Камера, Мпикс	Вспышка	Клавиатура	Цена, руб.*
Apple iPhone3GS	iPhone OS	320×480	135	256	3	✗	Экранная	29990 (32 ГБ)
Palm Pre	WebOS	320×480	135	256	3,2	✓	Аппаратная	21240 (8 ГБ)
HTC Legend	Android 2.1	320×480	126	384	5	✓	Экранная	18420 (без доп. памяти)
Nokia N900	Maemo	800×480	181	256	5	✓	Аппаратная	22230 (32 ГБ)

* Средняя цена под данным Яндекс.Маркет на июль 2010 года.

Программы

Маето

Nokia N900 работает под управлением Маето 5. На смену этой ОС идет MeeGo, совместная разработка Nokia/Intel, планировавшая достичь мощи Google и Apple. Однако MeeGo для N900 официально не поставляется, и мы остаемся с Маето на неопределенный срок. А когда MeeGo появится, то, по первым впечатлениям, она будет ближе к Moblin, чем к Маето. Это значит, что N900 – вероятно, последняя модель в линейке Nokia, которая работает под Linux на базе GTK.

Основная концепция Маето заимствована у виртуальных рабочих столов – только тут они называются не рабочими столами, а «видами». Шаркая пальцем по экрану влево или вправо, вы раскрываете разные виды, с прокручиванием значков и фонового изображения и эффектом параллакса. Каждый рабочий стол можно по выбору заполнить приложениями, ссылками и мини-приложениями – виджетами [widgets]. По умолчанию их немного: включены встроенная навигация, OVI-карты, ряд ссылок на сетевые ресурсы Nokia и социальные сети. Кроме того, есть кнопка менеджера окон, часы и индикаторы заряда батареи и уровня сигнала в левом верхнем углу экрана.

Тактильный интерфейс

Приложите палец к свободному месту любого из рабочих столов, и в верхнем правом углу появится небольшой инструмент настройки. Его можно использовать для входа в меню настройки или для удаления ссылок с экрана. Меню позволяет изменять тему оформления, фон, добавить закладки и добраться до виджетов: в стандартной установке они дают доступ к электронной почте, медиа-плееру, показу вашего текущего местоположения на карте и закладкам для доступа на Twitter и Facebook.

Для получения доступа к чему-либо еще коснитесь кнопки менеджера окон в левом верхнем углу экрана. Первое касание отобразит все запущенные приложения в миниатюре, и вы сможете активировать нужное «одним тычком». Коснувшись кнопки во второй раз, вы увидите прокручиваемый список установленных приложений: например, файловый менеджер, читалка PDF, календарь, инструмент для зарисовок, несколько игр и клиент обмена сообщениями через Skype, Google Talk, Facebook, Jabber и SIP-контакты. Есть даже X-терминал!

Значок, запускающий навигатор OVI-карт, изрядно напоминает значок web-браузера Safari, что способно вызвать путаницу. В ранних версиях Маето экраны с приложениями переключались кнопкой More [Еще], а другие экраны, например, экраны настроек, добывались вертикальной прокруткой. Иногда бывает сложно понять, как вернуться на предыдущий экран.

```

8:00 am X Terminal
BusyBox v1.10.2 (Debian 3:1.10.2.legal-losso30+0m5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ $ uname -a
BusyBox v1.10.2 (Debian 3:1.10.2.legal-losso30+0m5) multi-call binary

Usage: uname [-amnrspv]

~ $
  
```

» Как по-вашему, много ли на свете мобильных телефонов с предустановленным X-терминалом, готовым к работе?



Однако эти недостатки устраняются с каждым обновлением. Теперь кнопка More убрана из интерфейса, а сам интерфейс с каждым новым релизом работает быстрее.

Web-браузер Маето называется *MicroB*; он основан на *Firefox*, удобен и обеспечивает максимальную совместимость с сайтами. Но работает он медленнее, чем браузер на базе *WebKit*, да к тому же ему присущи некоторые раздражающие недостатки *Firefox*, в том числе – нудная процедура «игнорирования сертификата». Впрочем, в последних числах мая вышла новая прошивка, устранившая наше главное нарекание: теперь сайты можно просматривать в режиме «портрет», что было недоработкой прежних версий.

Лучшее в N900 – менеджер приложений, портал в мир ПО от третьих лиц. У N900 весьма рьяное сообщество энтузиастов-разработчиков, и немало их приложений заслуживают скачивания. Так, к ним относятся фантастическая игра *Angry Birds*

» Маето приносит мир социальных сетей прямо на ваш телефон – вы увидите ссылки на них на первом же экране.

«Лучшее в N900 – менеджер приложений, портал в мир ПО от третьих лиц.»

и даже *Brain Party*. Есть и клиентская и серверная части SSH, а также десятки других полезных инструментов, утилит и эмуляторов. Существует также версия *PyQt*, снабженная редактором, позволяющим писать собственные приложения, благодаря тому, что и *Qt* теперь является частью Маето. Эта ситуация – полная противоположность тому, что имеет место в iPhone, и основная причина, по которой стоит выбрать N900.

Android 2.1

Несмотря на ранний скептицизм, дела у платформы Android от Google идут неплохо. Недавний отчет от группы NPD намекает, что в США она скоро превзойдет по продажам iPhone. Это не такой уж сюрприз, если осознать, что существует множество других телефонов, использующих другие ОС, отличные от разработанных Apple. Но потенциально, Android – больше чем просто ОС для мобильного телефона. Уже сейчас Android осуществляет прорыв на рынок мобильных устройств, включая конкурентов iPad, недавно анонсированных Dell и HP, а также устройства, ориентированные на мультимедиа-информацию – например, два новых «домашних» планшета от Archos. Кроме того, за свою короткую историю эта ОС претерпела уже немало изменений. Большинство устройств сейчас используют версию 2.1, при этом версия 2.2 уже представлена разработчикам, выпущена для Nexus One и обещана для коммуникаторов HTC Desire и Legend в конце этого лета. Она

проста в использовании и заимствует лучшие идеи не только у iPhone и Maemo, но даже у старых версий Windows Mobile.

Android в оболочке

Наши устройства Legend используют Android 2.1. Как и у других телефонов от HTC, включая ориентированные на Windows Mobile, здесь есть индивидуальная графическая оболочка, работающая поверх стандартной ОС и прозванная в HTC 'HTC Sense'. Устройства HTC Desire и HTC Legend были первыми мобильными телефонами, получившими эту новую версию.

Первое, что вы заметите, включив устройство нажатием кнопки питания и проведя пальцем по экрану сверху вниз – это яркость дисплея и четкость изображения. Это – 'HTC Sense', перекидной календарь, часы, апплет погоды и еще несколько ссылок на наиболее широко применяемые функции устройства. Как

«Android Market предлагает десятки тысяч игр, утилит и приложений.»

и в N900, можно выполнять прокрутку влево или вправо, для доступа к семи рабочим столам. Каждый из них можно перенастроить на другой апплет погоды, меню SMS, электронную почту, быстрый доступ к избранным контактам и список закладок. На этот экран можно добавлять собственные виджеты (у вас будет из чего выбирать), в т.ч. имеющие доступ к вашим учетным записям в социальных и музыкальных сетях, YouTube и сетях фотосервиса.

Данные приложения предлагают сразу же видимый глазу сервис, но есть масса других, более продвинутых программ, доступных при нажатии на стрелку в левом нижнем углу экрана. Они выведут вас прямо в Android. По умолчанию, весь этот стандартный набор приложений такой же, как в Maemo на Palm Pre и N900.



► Разработка Android ведется уже несколько лет, но функция Google Search всегда была наготове.

Исключения, в основном, возникают вследствие того, что Android-устройство связано с вашей учетной записью Google. Например, при щелчке мышью на кнопке Talk запустится клиент обмена сообщениями Google Talk. Щелчок по кнопке Mail немедленно выведет вас на папку входящих сообщений вашего почтового ящика Gmail.

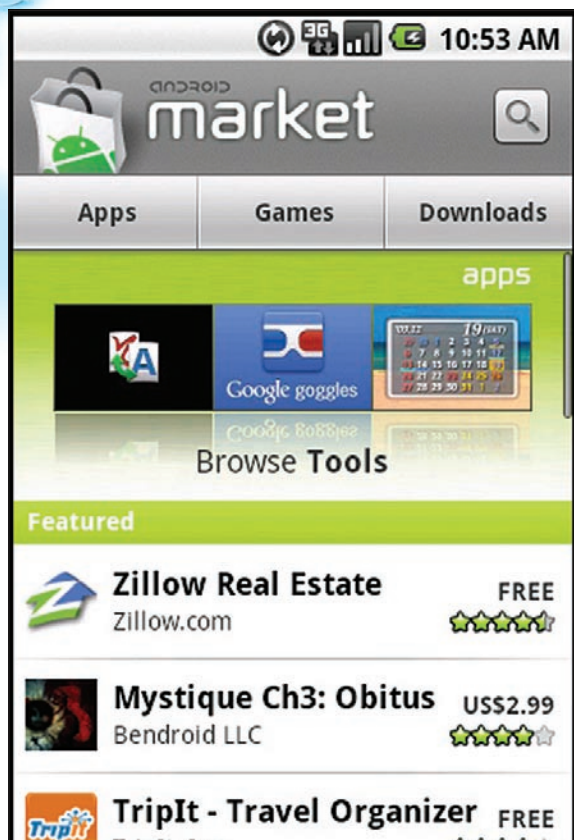
Google: и это все о нем...

Такая Google-центричность не распространяется, правда, на апплет электронной почты, который требует стандартных настроек POP3/IMAP или сервера Exchange, зато вы сразу видите входящие сообщения Gmail по мере их прибытия, а также свой список контактов и календарь, которые заполняются задаром и без дополнительных настроек Google-серверов. Приятное свойство календаря – создание «расписания» на много дней вперед, снабженное даже прогнозом погоды. Поиск тоже очень мощный, и, вполне ожидаемо, охватывает не только локальные, но и удаленные данные. Например, нажав кнопку с изображением увеличительного стекла и введя строку вроде 'Linux Format', вы увидите соответствующую запись в онлайн-режиме Google, а также в календарях, почте, заметках и приложениях, в том виде, в каком вы сохранили их на своем телефоне.

В принципе, по функциям Android не так уж сильно отличается от других ОС для мобильных телефонов. На нашем устройстве HTC Legend все работало стабильно и быстро. Хотя переходы и не так гладки, как у iPhone 3GS, Android-устройство сумело обогнать N900. Впрочем, «сердцем» этой ОС все равно оказалось приложение Market – портал, предназначенный для приложений от сторонних разработчиков. Приложение Android Market – главный конкурент Apple App Store; оно предлагает для загрузки на ваше устройство десятки тысяч игр, утилит и приложений. Его интерфейс весьма приятен, загрузка и установка приложений происходит в фоновом режиме, в отличие от необходимости



► Установщик приложений Android под названием Market – лучший из трех рассмотренных и единственный реальный конкурент для iStore от Apple.





Android 2.2

переключаться, как в iPhone. А лучшее свойство Market – то, что вы не ограничены приложениями, официально поддерживаемыми Google. Можно просто накидать сторонние приложения в формате APK-файлов на свою карту памяти, а потом установить их вручную через все то же приложение Market.

Но, возможно, они вам не потребуются. На конец апреля 2010 года портал Android Market насчитывал более 50000 приложений, тогда как Apple AppStore – более 200000. Разрыв выглядит значительным, но Android Market сейчас находится примерно на том же уровне развития, на котором AppStore был в июне 2009 года. При этом резкий рост числа приложений в AppStore не прогнозируется, поскольку политика публикации программ в нем становится все менее предсказуемой.

Web OS

Фирма Palm имеет долгую и хорошо продуманную традицию разработки ОС, эффективных и в плане ресурсов, и в плане удобства работы, и наиболее полно учитывающих и ваши технические возможности, и интуицию. В итоге, лучшее, что есть в WebOS от Palm Pre – это скорость. Это быстрая, чуткая и интуитивно понятная ОС, чуть более «гладкая», чем Android и Maemo. Она набирает популярность у пользователей, далеких от техники, поскольку не топтит основные функции в море доступных опций и приложений. Кроме того, она отлично эксплуатирует сенсорный экран, позволяя пользователям выделять, масштабировать и вращать интерфейсные элементы практически так же, как в iPhone.

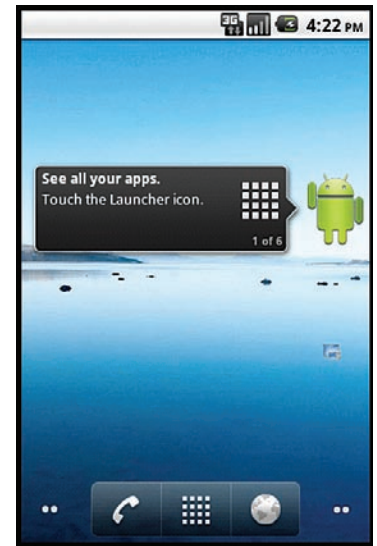
Основной экран выглядит как слегка усохший экран iPhone с пятью кнопками быстрого запуска на нижней границе. Главный экран допускает прокрутку списка всех установленных приложений и утилит влево и вправо. Нажмите единственную кнопку перед экраном – и развернутое в настоящий момент во весь экран приложение исчезнет в фоне, демонстрируя фоновый рисунок и программы, расположенные справа и слева. На устройстве Palm Pre Plus аппаратная кнопка удалена, и для того же эффекта нужно просто провести по экрану пальцем. Двигая пальцем влево или вправо, можно переключаться между приложениями или вернуться в главное меню, чтобы запустить что-либо еще. Для

Релиз Android 2.2 – масштабное обновление системы. Сейчас он доступен для Nexus One и в виде SDK для разработчиков, а когда вы будете читать этот материал, уже доберется и до Legend и Desire. Помимо ошеломляющего повышения скорости, список улучшений включает превращение вашего мобильного устройства в точку доступа Wi-Fi, способную связать с мобильной сетью ваш ноутбук или игровую приставку.

Второе серьезное изменение – способ представления вашей фотоколлекции. Похоже, разработчики новой версии черпали вдохновение у VimpTop, компании-создателя тактильных интерфейсов, приобретенной Google ранее в этом году. Новый интерфейс позволяет визуально группировать фотографии по пачкам и раскладывать их, используя жесты. В изобилии представлены различные элементы управления камерой в процессе съемки.

Однако то, что рекламировалось как основное новшество, оказалось как раз разочарованием – а именно, внедрение Flash-технологий на Android Market. Производительность нынешней реализации просто ужасна. Хотя вы и получаете доступ к большинству Flash-контента, просмотр страниц и сам браузер Android (в остальном превосходный) сильно тормозят. Уж лучше пользоваться сайтами, оптимизированными для мобильного доступа,

а не полагаться на технологии, которые зачастую ставят на колени и полномочный Linux-ПК.



➤ С виду релиз Android 2.2 похож на предшественников, но способен изменить ситуацию на игровом поле.

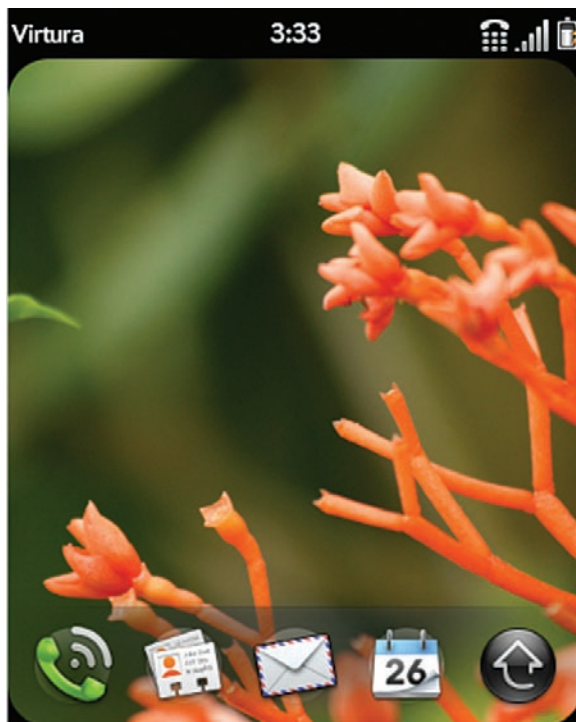
закрытия приложения достаточно «мазнуть» по нему пальцем по направлению вверх.

По умолчанию вы получите версию Google Maps, учетную запись YouTube, весьма похожую на ту, что есть у iPhone, медиаплеер и фотобраузер. Имеются также программы для чтения текстов и PDF-файлов. Как и в N900, синхронизация данных с рабочим столом не требуется, да и способа для нее нет. К счастью, многие вещи можно сохранять и забирать через «облако» или обращаться к ним из браузера.

Просмотр Web

Web-браузер в этой ОС великолепен. Основанный на WebKit, он миглом загружает страницы, а Multitouch-интерфейс облегчает навигацию. Браузер не умеет создавать новые вкладки, но многозадачная природа WebOS позволяет тут же запустить еще один экземпляр браузера, и он будет работать параллельно текущему сеансу. Поддержки Flash не предвидится, но это – общая глобальная проблема.

Возможно, слабейшая сторона WebOS – нехватка сторонних приложений. Портал для них имеется, так же, как у Nokia и Android, но платформа не смогла привлечь сторонних разработчиков и убедить их портировать приложения под эту систему. По сравнению с конкурентами, платформа занимает третье место. Это – самый серьезный недостаток устройства: вы не сможете надставить его дополнительными приложениями и утилитами. Быстрый рост числа мобильных приложений – основное отличие современного поколения смартфонов от предшествующего. Пока Palm Pre в этой области уступает соперникам. Впрочем, есть надежда, что за счет инвестиций нового владельца Palm, корпорации HP, появятся новые мобильные устройства под WebOS, выйдет ее новая версия, и разработка дополнительных приложений оживится. В противном случае смена владельца может стать «последним гвоздем в крышке гроба» устройств Palm.



➤ Каждый из «видов» WebOS можно заполнить собственным набором виджетов и ссылок.

Под капотом

Как экспериментировать с приложениями и разрабатывать свои.

Каждая из трех описанных платформ построена на Linux, и, в отличие от ряда конкурентов, с ними можно экспериментировать, используя Linux, а также разрабатывать для них собственные приложения и распространять их среди пользователей.

Каждая из платформ успешно конкурирует со средой разработки Apple, и рассмотренные здесь устройства во многом превосходят продукцию Apple. Например, вы не обязаны использовать только Objective C или платить за лицензию на разработку ПО. Вы также не привязаны только к официальным API и к невнятным руководствам разработчика.

Android

Разработка приложений для Android очень проста, особенно по сравнению с закрытой системой разработки Apple, которая «замкнута сама на себя». Скачать и установить Android SDK может каждый. Начните с запуска небольшого вспомогательного пакета, и когда он отработает, все необходимое, включая основную среду разработки приложений, будет скачано и установлено в ваш домашний каталог.

С помощью SDK можно запустить виртуальную модель телефона Android, даже не связываясь с исходным кодом. Просто запустите Android SDK и AVD Manager, щелкните по кнопке New – и готово: играйте себе с виртуальной версией Android. В ней только не будет приложения Market. Из среды разработчика вы сможете отправлять свои приложения прямо в работающую систему Android и отлаживать их точно так же, как на реальном устройстве.

Можно создать виртуальное устройство и отправлять на него свои приложения через модули расширения, установленные в вашей IDE. Как и для устройств Palm, большинство разработчиков предпочитают среду Eclipse, и модули, призванные упростить ситуацию, входят в загружаемый пакет. Если вы заинтересовались разработкой для Android, обратитесь к нашим учебникам по ней, опубликованным несколько месяцев назад.



► SDK Android – лучший из всех рассмотренных платформ: он включает готовый эмулятор телефона.

Желающие «засесть» на телефоне Android точно так же, как на компьютере с Linux, не обойдутся без проблем. Как и в любом из этих устройств, ключевые компоненты «заблокированы» в зашифрованной ROM, и пользователи не могут кардинально изменить конфигурацию системы и аппаратных средств. Но для любой схемы защиты существуют обходные пути, и многие взламывают системы Android ради полного доступа к аппаратным средствам.

WebOS

Как и Android, Palm предоставляет виртуальную версию своей ОС для мобильного телефона, и в ней можно экспериментировать и вести разработку приложений. Но она и вправду виртуальная, и процедура установки требует, чтобы вы, наряду с различными элементами SDK, раздобыли и новейшую версию VirtualBox. Заставить всю эту комбинацию работать несколько сложнее, чем единое главное приложение Android, но не убийственно сложно.



Лучшие приложения

Каждое устройство имеет свой собственный портал приложений, где можно выбирать новые программы для установки на ваш телефон. Как правило, это делается в одно или два касания. Для каждого устройства доступны бесконечные списки приложений, и мы сконцентрируемся на наиболее интересных большинству линуксоидов. Лично нам больше всего нравится SSH. Android – единственная платформа, поддерживающая SSH через портал приложений, где предлагаются и клиентская, и серверная части. Но существуют и сторонние приложения для N900, которые тоже не вызывают особых проблем. Осложнения есть только у Palm Pre.

Наше следующее требование – файловый менеджер. Для Nokia N900 он входит в комплект поставки, и это впечатляет. Однако Android предлагает аж семь вариантов, и все бесплатные. Мы предпочитаем файловый менеджер Apollo. Поработав со всеми тремя телефонами, вы скоро оцените широту выбора Android Market. По-

иск для Linux на N900 не дает ни одного результата, а для Android – целых 64, включая массу книг O'Reilly на тему Linux и даже приложение, которое предложит вам установить Ubuntu на устройстве Android. Предлагаются также клиент и сервер VNC, пульт удаленного управления медиа-плеером Banshee, а также виджеты, утилиты и игры, позаимствованные практически из всех свободных проектов.

Android для приложений

С коммерческими приложениями ситуация та же. Есть несколько отличных игр для N900, в первую очередь – Angry Birds и Airport Touch, а также приложения наподобие Firefox и картографического приложения Nokia. Но это и все. Для Android выбор намного шире, включая официальное приложение Twitter и поддержку сервисов социальных сетей типа Facebook и Four Square. Есть приложение Google Goggles, выполняющее поиск по изображениям, а также лучшие Linux-версии карто-

графического ПО и сотни прочих приложений, в основном бесплатных. Сервис приложений Android приближается к iPhone, и по этой причине любителям скачивания следует предпочесть Android.



► Дешевое приложение от третьих лиц превратит ваш телефон в мини-пианино.

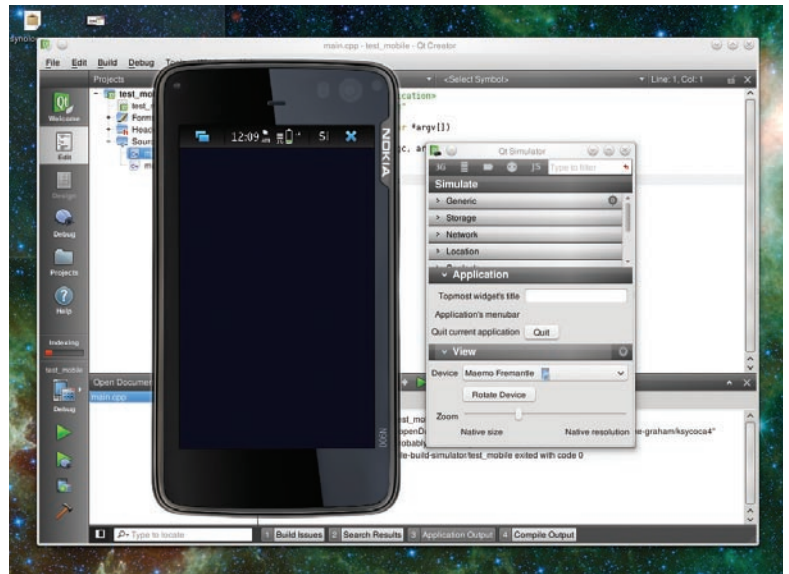
Запустив эмулятор, вы сможете экспериментировать с WebOS на своем компьютере и без реального телефона. Единственное отличие в операционной системе – то, что, как и в Android, в виртуальной среде нет портала приложений. Опять же, как и в случае с Android, в данном случае доступны модули расширения для среды разработки *Eclipse*, позволяющие поддерживать коммуникации с вашим устройством и его виртуальной версией.

Несмотря на миллионные продажи, WebOS отчаянно нуждается в новом ПО, способном дать разработчикам преимущество над Android Market, и ваши приложения имеют больше шансов быть замеченными. Далее, обновление и самой ОС, и ее SDK в начале года, вероятно, намного упростит разработку: включена *SDL*, группа открытых библиотек, часто применяемая для разработки игр и эмуляторов. А значит, другие инструменты *SDL*, которые тысячами пасутся в типовом менеджере пакетов Linux, тоже довольно легко портировать на WebOS, особенно если сравнить это с трудозатратами при начале с нуля или с изучения языка *Objective C*.

Однако на самом низком уровне *Palm Pre* имеет еще одну общую черту с Android – а именно, блокировку на уровне ядра. В итоге вы не получите полного контроля над устройством без выполнения неофициальной и довольно загадочной процедуры.

Маето и MeeGo

И Маето, и MeeGo находятся в переходном состоянии. Хуже того, обновление N900 для MeeGo не планируется, а это значит, что разработку приложений для Маето 5 можно оспорить как пустую трату времени. Но если вы думаете о дальних перспективах, то Linux – прекрасная платформа для разработки под MeeGo. Дело в том, что официальная среда разработки для MeeGo – *Qt Creator*, и она намного проще в освоении, чем *Eclipse*. Поскольку владельцем и разработчиком *Qt Creator* является Nokia, вам достанутся множество примеров, хорошая документация и платформа, полностью поддерживаемая компанией-изготовителем аппаратных средств. Так что разработка под MeeGo для будущих устройств Nokia и Intel будет во многом схожа с аналогичным опытом для



устройств Apple, особенно в комбинации с SDK *Qt* от Nokia. Это отдельный пакет, объединяющий *Qt Creator* и все необходимые библиотеки разработки, а также эмулятор, способный в полном объеме имитировать ваш телефон.

► SDK от Nokia – не полный эмулятор, но включает среду разработки.

Linux против iPhone: победитель

Мобильный Linux – это беспрецедентный успех. На рынке, где годами доминировали Nokia, Microsoft и им подобные, честью для нашей любимой ОС является то, что она быстро адаптировалась и заняла свою нишу в «экосистеме» мобильных устройств за столь короткий срок. Поразительно и то, что открытая система успешно конкурирует с Apple, причем не требуя колоссальных бюджетов на НИР и разработку ПО, не закликаясь на единственной точке зрения и не ограничивая свободу пользователей, хотя и не без помощи от Google и подобных компаний.

Наиболее впечатляет то, что мобильные телефоны на базе Linux превосходят iPhone не только в идеологии свободы ПО. Во многих случаях они просто лучше. Владельцам iPhone практически недоступны такие простые вещи, как изменение «домашнего» экрана или браузеров музыки и фото. А во всех трех рассмотренных платформах это до смешного просто. Их API не монополизируют-

«Мобильные телефоны на Linux превосходят iPhone не только в идеологии свободы ПО.»

ся единственным разработчиком и не налагают «драконовских» ограничений на использование, и вы обладаете полной свободой выбора при установке и разработке любых приложений, невзирая на взгляды разработчиков.

Но главным доводом в их пользу является то, что все они работают под Linux, и хотя и нельзя так глубоко проникнуть в ядро ОС, как на настольном компьютере, это тоже преодолимо. Многие инструменты и приложения Linux уже портированы на эти устройства, а на порталах приложений имеется множество приложений от сторонних производителей, разработанных на основе открытых проектов. Это значит, что вы, возможно, уже знакомы с ними, а также то, что эти телефоны будут долговечными. Аппаратные средства меняются; то же происходит с операционной системой и API, но самый фундамент свободного ПО остается незыблемым и со временем будет только укрепляться. LXF



► Вы не получите особого контроля над устройством, но Palm предлагает версию своей ОС, работающую в *VirtualBox*.

Архитектор

Джоно Бэкон беседует с Дэвидом Зигелем — человеком, который хочет осчастливить вас через Ubuntu.



Амбиции Ubuntu не ограничиваются перемещением кнопок в окне и тотальным окрашиванием в пурпур — на недавнем саммите разработчиков Ubuntu в Бельгии (Ubuntu Developer Summit) было объявлено о Unity, совершенно новом интерфейсе рабочего стола, нацеленном на технологию мгновенной готовности (instant-on). Нас весьма заинтриговал тот факт, что вместе с командой проектировщиков трудится создатель фантастического *Gnome Do*, Дэвид Зигель [David Siegel]. Естественно, мы захотели выяснить как можно больше...

Linux Format: Как вы попали в мир Open Source?

Дэвид Зигель: Мне нужно было создать оболочку Unix и Unix-подобное ядро для университетского курса по разработке операционных систем. Столкнувшись с несовместимостью системных вызовов Mac OS X и лабораторных компьютеров с Linux, я установил Ubuntu на свой Mac, чтобы привести мою среду разработки в соответствие среде, имевшейся на лабораторных компьютерах. И в конце этого процесса я подумал: «В этом Ubuntu есть все, что надо; может, и дальше работать с ним?»

Следующим летом я работал в Google с Шоном Иганом [Sean Egan], который тогда был куратором *Pidgin*. Шон рассказал мне, каково это — работать над открытым проектом, и его обязанности меня впечатлили.

Моему переходу с Mac OS X на Ubuntu препятствовало только *Quicksilver*, клавиатурное приложение от Николаса Житкофф [Nicholas Jitkoff]. Для выпускной работы по компьютерным наукам я создал проект программы с открытым кодом для исследования функции поиска в среде рабочего стола, с конечной целью создать подобное приложение для Linux; результатом стал *Gnome Do*.

LXF: Как вы оказались в Canonical?

ДЗ: Я встретил Марка Шаттлворта [Mark Shuttleworth] в Бостоне на Gnome Summit 2008 и поделился с ним своими идеями об опыте пользователей и о свободном ПО. Он предложил мне зайти в лондонский офис Canonical на собеседование и подумать, не заинтересует ли меня переезд в Лондон для работы в создающейся команде разработки Canonical. По случайному совпадению, я как раз планировал ехать в Лондон на следующей неделе, вот и зашел. Сначала я решил не работать с Canonical, но через восемь месяцев Марк пригласил меня на соревнования по разработке в Кейптауне, и я был так восхищен, что не смог отказаться!

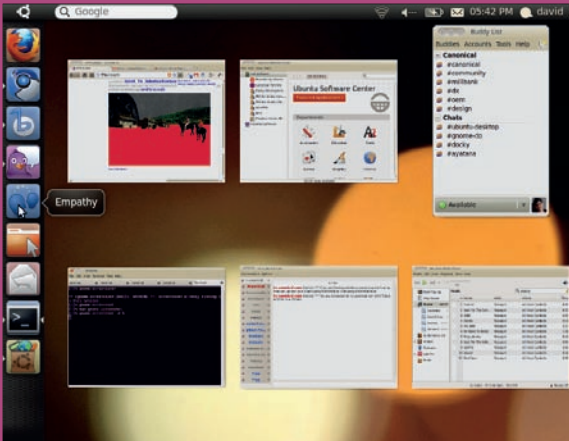
LXF: Где ваше место в команде разработки и в видении Ubuntu?

ДЗ: Моя роль в команде разработки — «Архитектор взаимодействия со средой рабочего стола (Desktop Interaction Architect)». Я пишу примечания и создаю модели, описывающие использование Ubuntu. Другие члены команды разработки превращают эти описания в интерактивные прототипы и визуальные образы, которые могут быть протестированы пользователями и впоследствии реализованы.

LXF: Вы работаете с более широким сообществом Ubuntu?

ДЗ: Когда я не создаю «архитектуру взаимодействия со средой рабочего стола», я пытаюсь породить интерес к опыту пользователей в сообществе свободного ПО. В этой связи я возглавляю проект One Hundred Paper Cuts и недавно анонсированный проект UX Advocate, работая в Canonical.

Что касается «виденья дизайна» Ubuntu, я не могу сказать с уверенностью, какую роль я играю в нем, но я пытаюсь заставить тех, кто занимается технической стороной Ubuntu



› Интерфейс Unity — это часть массовых улучшений, сгруппированных в проект *Anataya* от Canonical.

смотреть на программы не только как на возможность написать красивый исходный код, но и как на возможность создать прекрасный инструмент для людей.

LXF: И этим вы собираетесь заниматься с помощью Unity?

ДЗ: Unity — это легковесный интерфейс для нетбуков Ubuntu. В более широком смысле, это оболочка, созданная для устройств с маленьким экраном. Unity объединяет программу запуска, которая превращает процесс открывания приложений и переключения между ними в настоящее удовольствие; панель, где индикаторы ведут себя одинаково; просмотр установленных приложений с интеграцией с Ubuntu Software Centre; просмотр файлов с быстрым доступом к любимым папкам, недавно созданным файлам, загрузкам и возможностью простого просмотра; поисковый интерфейс, позволяющий повсеместно использовать поиск по мере набора, так что можно находить файлы, приложения и настройки несколькими нажатиями на клавиши.

LXF: Каково ваше видение Unity? Чего вы хотите достичь с его помощью?

ДЗ: Недавно Canonical объявила об Ubuntu Light, версии Ubuntu с интерфейсом, нацеленным на создание возможности быстрой, простой и безопасной работы в Web. Ubuntu Light и Ubuntu Netbook Edition во многом пересекаются, главным образом потому, что оба оптимизированы под маленькие экраны и веб-серфинг. Unity служит основой обоих продуктов, и они могут делить между собой общие элементы, такие как индикаторы и программа запуска. Моя краткосрочная цель в Unity — создать фантастический опыт для Ubuntu Netbook Edition 10.10, расширив Ubuntu Light до поддержки основных приложений и возможностей управления файлами, подходящих для устройств общего назначения типа нетбуков.

Хотелось бы и дальше задействовать поиск и реализовать тактильный интерфейс. Мне интересно использовать поиск для решения сложных задач, поставленных пользователями (однако я не рассматриваю поиск как панацею), и все в наше время интересуются сенсорными устройствами.

LXF: И когда Unity достигнет уровня версии для нетбука?

ДЗ: Надеюсь, он выйдет в Ubuntu Netbook Edition 10.10! До октября нам предстоит много работы, но, поскольку

О РАБОЧЕМ СТОЛЕ UNITY

«Unity — это легковесный интерфейс для нетбуков с Ubuntu.»

во главе инженерных работ Unity стоит Нейл Пейтел [Neil Patel], я спокоен.

LXF: Как по-вашему, улучшит ли Unity состояние имеющихся на сегодня интерфейсов для нетбуков?

ДЗ: Я уже говорил, что работа с программой запуска Unity — это истинное удовольствие, и я не шутил. Первая версия программы запуска просто прокручивала экран вниз по мере его заполнения. Вторая версия, готовая к отгрузке с Ubuntu 10.10, ведет себя совершенно иначе. Словами этого не опишешь — просто дух захватывает. Если кто-то увидит, как вы работаете с Unity, вас непременно спросят: «Ух ты, это что такое?». Это — не просто улучшение существующих интерфейсов для нетбуков, это — улучшение интерфейсов персональных компьютеров в целом.

LXF: Велись определенные дискуссии по поводу Приложений и Файлов. Что это такое?

ДЗ: Приложения [Applications Place] — это отображение в Unity установленных у вас приложений. Оно позволяет просматривать установленные у вас приложения и позволяет осуществлять поиск по мере набора как среди установленных приложений, так и среди приложений, имеющихся в Ubuntu Software Centre. Оно просто превосходно.

Файлы [Files Place], отображение в Unity файлов на вашем нетбуке, отказывается от традиционной навигации по иерархической файловой системе, предлагая вместо нее поиск и просмотр на основе времени. Все это превращает Ubuntu Netbook Edition в первый интерфейс для нетбуков с просмотром файлов на *Zeitgeist* [новый файловый менеджер в Gnome 3]. Еще многое надо сделать, но и это — уже дерзкий и волнующий эксперимент, и он бросает вызов многим привычным представлениям о том, как люди думают о своих файлах.

LXF: Некоторые обвинили Ubuntu в мимикрии под Mac OS X — что вы об этом думаете?

ДЗ: Я не занимаю никаких позиций по поводу всякой ерунды, которую говорят люди, но если мне пришлось бы какую-то занять, то я сказал бы: злопыхатели всегда будут злобствовать.

LXF: Что вы считаете конечной целью и возможностью настольного Ubuntu?

ДЗ: Самая большая возможность для Ubuntu — принести людям радость, а его цель — добиться этого, не нарушая этики, то есть не рассматривая своих пользователей как средство для достижения конечного результата; пользователей Ubuntu порадует отличный опыт работы, и если не требовать от них, чтобы они пожертвовали свободой, чтобы использовать программу, то это и будет достойным отношением к ним.

LXF: Как можно принять участие в Unity?

ДЗ: Unity доступен через PPA `ppa:canonical-dx-team/une`. Добавив этот PPA в свою систему Ubuntu (сейчас это должен быть Lucid), установите пакет `unity`, затем поменяйте сессию с Gnome на Unity в окне приглашения. Установив Unity, пожалуйста, поиграйте с ним, чтобы сообщить об ошибках. Вы найдете меня на канале IRC `#ayatana` на `irc.freenode.net`, мой ник там — 'djsiegel'. Загляните туда пообщаться со мной о Unity; я жажду узнать о ваших впечатлениях и предложениях! **LXF**



Fedora и RHEL Тандем победителей

При грядущем релизе Red Hat Enterprise Linux и успехе Fedora 13, у Red Hat есть чем козырнуть перед Шашанком Шармой.



Корпоративная поддержка и большое активное сообщество — практически все дистрибутивы Linux могут похвастаться хотя бы одним из двух. Fedora, с момента своего создания в конце 2003 года как открытого дистрибутива Red Hat, вырастила вокруг себя преданное сообщество, релиз за релизом выпуская инновационные и полные дистрибутивы, заслуживающие внимания и уважения.

Будучи довольно большим дистрибутивом (число DVD-дистрибутивов, по сравнению с распространяющимися на единственном CD, не так уж и велико), Fedora 13 позволяет любому линуксоиду найти что-нибудь для себя. Давайте возьмем свежую Fedora 13 и почти готовый RHEL 6 и рассмотрим этих двух монстров свободного ПО — выясним, что предлагает Fedora 13 и почему настало время для RHEL 6.

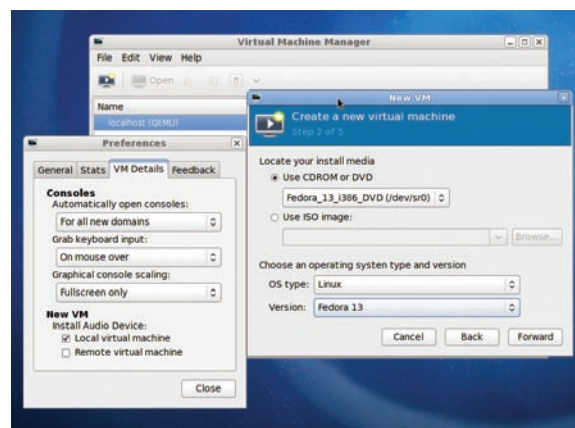
f Откровения

По словам Пола В Фрилдса [Paul W. Fields], руководителя проекта Fedora, процесс создания функций для Fedora и пестрое сообщество разработчиков и участников позволяют охватить широкий спектр функций в каждом релизе. «Fedora 13 включает немало настольных аспектов, помогающих любому пользователю повысить эффективность оборудования — от поддержки 3D для видеокарт до управления цветом для устройств ввода и вывода или автоматической установки драйверов принтера. Но этот релиз также принес продвинутые функции для разработчиков, такие как улучшение инструментов мониторинга, позволяющих Python-программисту произвести оценку деятельности его системы и найти узкие места в создаваемом коде Python. А системные

администраторы восхитятся редизайном инструмента аутентификации в Fedora 13 с использованием системы System Security Services Daemon (SSSD) для обеспечения управляемого входа в домен, даже для пользователей ноутбуков, находящихся вдали от сети».

f Лидер виртуализации

Fedora стабильно являлась плацдармом для технологий виртуализации, и версия 13 продолжает эту тенденцию. Охвачены передовые технологии, выводящие виртуализацию на новый уро-



» При создании новой виртуальной машины аккуратно выбирайте тип ОС и версию.

вень. Фрилдс говорит: «Как всегда, Fedora продолжает лидировать в области виртуализации, так как разработчики нашего сообщества на самом деле активно участвуют в таких передовых областях, как разработка ядра и гипервизора KVM». Хотя Fedora несколько лет придерживалась Xel, затраты времени и энергии, потребные для того, чтобы заставить его работать с ядром Linux, были явным недостатком. Поддержка стабильной PCI-адресации в KVM и Virt Shared Network (разделяемая сеть) стали двумя основными новшествами KVM в Fedora 13. Технология Shared Network Interface позволяет виртуальным машинам использовать те же физические сетевые карты (NIC), что и основная ОС.

Все виртуальные машины под Fedora 13 управляются посредством инструмента Virtual Machine Manager в Приложения > Служебные. Вы можете создать или восстановить существующие виртуальные машины за несколько минут, ведь интерфейс пользователя очень прост. Многие недавние релизы дистрибутивов требуют как минимум 1 ГБ ОЗУ, и если вы не располагаете таким объемом, то, вероятно, не сможете запустить во вновь созданной виртуальной машине графическую ОС.

Отлаженная установка

Fedora никогда не была слишком сложна в установке. Тем не менее, Fedora 13 поставляется с более продвинутой версией инсталлятора Anaconda, еще упростившей этот процесс, благодаря улучшениям в работе с дисками и разбиении на разделы. Теперь можно также установить Fedora через Интернет.

Загрузочные образы доступны для различных носителей, включая USB и CD на boot.fedoraproject.org. Такие образы позволяют подключаться к системе на удаленном сервере для запуска установки, избавляя от необходимости иметь 700-МБ CD или 4-ГБ DVD.

Поддержка оборудования

Застарелым аргументов против Linux было то, что здесь нет такого уровня поддержки аппаратуры, как у проприетарных операционных систем. По этой части, Fedora 13 предлагает драйверы Nouveau с экспериментальной поддержкой 3D для видеокарт Nvidia, и пользователям больше не придется полагаться на ненадежные проприетарные драйверы, которые нельзя отладить и улучшить. Реальным достижением, однако, является функция Automatic Printer Driver Installation.

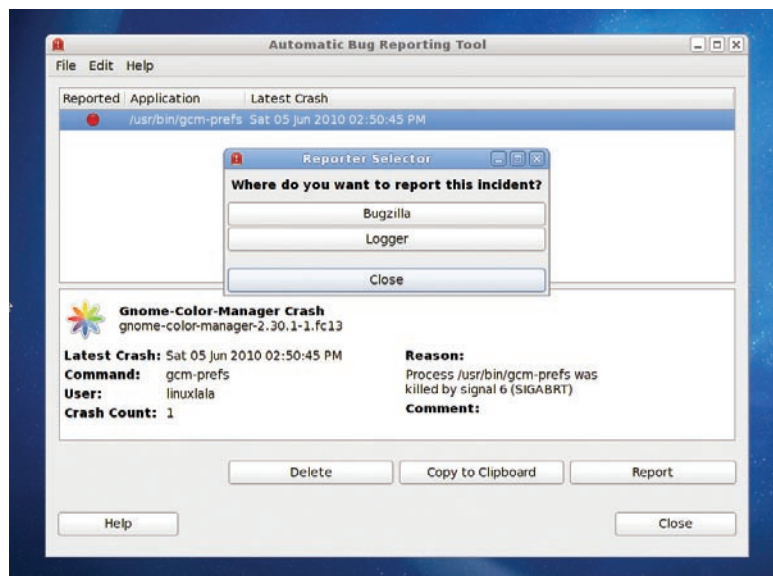
Все принтеры, будь то подключаемые через USB, параллельный порт или по сети, определяются своей строкой Device ID, содержащей информацию об изготовителе, названии модели, поддерживаемом наборе команд и т.п. Исторически, настройка принтера была обременительна для большинства пользователей – нередко из-за незнания должного драйвера.

Представьте себе, однако, что драйверы принтера содержат теги, связывающие их с определенными изготовителями и номерами модели, так что Fedora, обнаружив принтер, сразу же ищет драйвер, соответствующий меткам изготовителя и модели, и автоматически устанавливает его. Теперь это возможно в Fedora 13, отсюда и такое заманчивое название функции.

Соратник программиста

Предоставляя возможность параллельной установки Python 3, то есть Python 3.1.2 теперь может работать бок о бок с Python 2.6.4, Fedora 13 позиционирует себя как идеальную платформу для разработчиков.

Python 3 решил многие давно назревшие проблемы Python 2, но при этом превратился почти что в другой язык. Инструмен-



том *2to3* в Python можно автоматически преобразовать значительную часть кода Python 2 в Python 3, но тут есть ловушка. Говоря Python, мы имеем в виду три взаимосвязанных компонента: среду времени выполнения, стандартную библиотеку и множество сторонних модулей. Беда в том, что не все модули (а их сотни) были полностью перенесены на Python 3. И Fedora 13 обеспечивает разработчикам возможность интегрировать Python 2 и Python 3 для продолжения их работы, а также подготовить переход на Python 3.

Вторая функция Python позволяет разработчикам найти узкие места в коде Python, который они создают. *SystemTap*, инструмент отслеживания/зондирования/мониторинга, позволит пользователям и разработчикам следить за системой за пределами ядра. По существу, вы можете наблюдать, что происходит внутри вашего приложения и среды выполнения языка типа Python и т.п.

Поскольку код Python легко сочетать с кодом на других языках (например, C), третьим нововведением Python является расширенный *GDB* (GNU Debugger), который сообщает сведения о таких процессах и на уровне C, и на уровне Python. Вам не нужно быть экспертом *GDB* для отладки кода на Python: улучшенный *GDB* понятен даже зеленым новичкам.

Снимки файловой системы btrfs

Вы когда-нибудь боялись вступать в авантюры в вашей системе из-за того, чтобы не остаться с ломом вместо машины? Btrfs умеет создавать легковесный загрузаемый снимок файловой системы. Система отката с использованием btrfs позволит администраторам и пользователям вернуться к предыдущему образу, если в системе возникли ошибки. Btrfs снимает образы целой файловой системы, которые могут быть созданы вручную или автоматически по требованию пользователя, и вся файловая система вернется в свое прежнее состояние, когда вы вернетесь к предыдущему снимку. Например, если вы делаете снимок каждый раз при удалении или установке новых пакетов, возврат к предыдущему снимку не просто повлияет на состояние этих пакетов – он также затронет ваш домашний каталог, находящийся на разделе btrfs. Файловой системой по умолчанию в Fedora 13 является ext4, но вы можете легко выбрать btrfs во время процесса установки.

➤ Инструмент Automatic Bug Reporting позволяет создавать отчеты об ошибках даже новичкам. Он доступен из меню Приложения > Системные утилиты.

«Fedora 13 идет с продвинутой версией установщика Anaconda.»

»

История инноваций

Важным аспектом релиз-цикла Fedora является последовательное развитие ключевых свойств на протяжении нескольких релизов. Мы видели это на примере скорости загрузки: запуск Fedora 10 длился 30 секунд, превратившись в 20 секунд в Fedora 11. Это одно из свойств, над которыми будет продолжаться работа и в Fedora 14.

Аналогично, *Archaer*, ответвление *GDB* с улучшенной поддержкой C++ и скриптов Python, дебютировал в Fedora 11, а сейчас в Fedora 13 мы видим более совершенный *GDB*, который порадует любого Python-программиста.

«На протяжении многих выпусков мы опираемся на прочную основу инженерного опыта и работаем над расширением функциональности абсолютно свободной и открытой программной платформы, — объясняет Пол Фрилдс. — Возьмем, к примеру, свободные видеодрайверы. В Fedora 10 мы реализовали установку режимов в ядре (kernel modesetting) для ускорения загрузки на некоторых видеокартах ATI.

В Fedora 11 мы расширили эту функцию на большее число видеокарт и начали расширять поддержку 3D-ускорения в совершенно свободных драйверах видеокарт Intel. В Fedora 12 мы собрали драйвер на этой платформе, с экспериментальной поддержкой 3D для видеокарт ATI в драйвере 'radeon', а Fedora 13 включает не только стабилизацию поддержки ATI, но и 3D-расширение для карт Nvidia с помощью драйвера 'nouveau'».

Менеджер сети

Примером долгосрочного проекта является Network Manager, начатый Red Hat еще в 2004 году. К появлению Fedora 12, он стал решением *де-факто* по настройке сети почти во всех дистрибутивах. В Fedora 12, Network Manager представил поддержку широкополосной мобильной связи, и, наконец, в Fedora 13 мы получаем поддержку телефонных модемов для более старых моделей телефонов с Bluetooth. Предоставляется также интерфейс командной строки, позволяющий работать с Network Manager даже пользователям чисто текстовых систем.

Еще одна новая функция Fedora 13 — управление цветом. Пользователям дается возможность создавать уникальные цветовые профили для различных аппаратных устройств, таких как принтеры, сканеры и мониторы, что позволит художникам, фотографам, дизайнерам улучшить жизнь, используя свободное ПО.

Согласно Фрилдсу, достижения в области видео в значительной степени обеспечены инженерами, нанятыми Red Hat для расширения возможностей свободного ПО на рабочем столе. «История свободного видеодрайвера — лишь один пример совместной работы проекта Fedora и Red Hat, не только для интеграции, но и для улучшения свободного и открытого ПО».

Инструменты виртуализации

KVM должен войти в грядущий RHEL 6, и, как объясняет Фрилдс, это реальный пример нередкого объединения команд двух дистрибутивов. «Fedora — свободный дистрибутив, проект сообщества и исток для Red Hat Enterprise Linux... [он] служит общественной лабораторией по НИР. Fedora — система общего назначения, которая дает Red Hat и остальным членам сообщества шанс быстро вводить новейшие технологии».

Люди четко увидят отражение недавних и прошлых релизов Fedora в RHEL 6. В некотором смысле, глядя на релизы Fedora, можно сделать весьма точный прогноз технологических новшеств и функций следующего релиза Red Hat Enterprise Linux.

Тим Берк [Tim Burke], вице-президент по инжинирингу Linux в Red Hat, уточняет, что отдельные лица и предприятия часто готовы принимать участие в разработке Fedora, чтобы впоследствии некоторые функции внедрились в RHEL. «Мы все чаще видим клиентов с частными потребностями, готовых внести свой вклад в Fedora в интересах получения нужной им функции в Red Hat Enterprise Linux».

А так как все, от домовладельцев до изготовителей оборудования, заинтересованы в энергоэффективных системах, Берк развивает свой пример: «Многие люди, от конечных пользователей до изготовителей оборудования и госзаказчиков, проявляют интерес к энергоэффективности. Пользователи с обоих этих полюсов работали с нами над Fedora 12 и Fedora 13, проводя аудит и улучшая стандартные системные сервисы, чтобы уменьшить энергозатраты. Такой вид работы будет непосредственно применяться на постоянной основе в Red Hat Enterprise Linux 6».

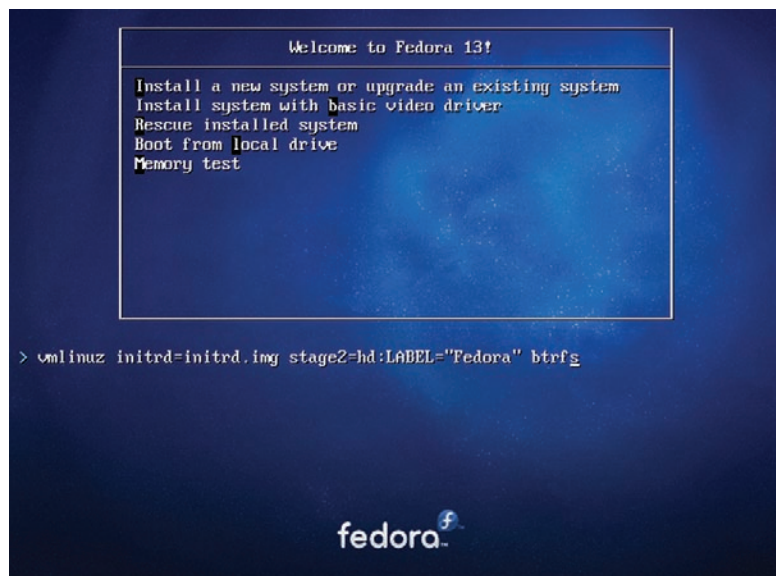
А в ближайшее время, RHEL...

Red Hat, кажется, был всегда, и успел пустить корни, где только можно, но все же избежал непопулярных сделок с гигантами проприетарного программного обеспечения, поиска вариантов для поглощения и дешевых выпадов против конкурентов. Как Red Hat держит свою линию?

По словам Берка: «В связи с тем, что вклад Red Hat является определяющим в большинстве ведущих технологий в составе ОС (от ядра, файловой системы, управления хранением данных, компиляторов, средств безопасности, до рабочего стола Gnome и настольных приложений), мы одновременно и производитель, и потребитель открытого исходного кода. Многие другие компании Linux больше потребляют, чем вкладывают в Linux, что смещает их позиционирование на рынке».

Одна из главных особенностей RHEL 6 — согласно Берку, постоянное снижение совокупной стоимости развертывания дата-

«RHEL 5 ввел Xen как свой выбор платформы визуализации.»



► Вы не сможете воспользоваться файловой системой btrfs, пока не загрузите установочный диск с опцией btrfs.

центра. Еще один ключевой момент – виртуализация. Недавние выпуски Intel и AMD вложили новый смысл в понятие «массовый процессор». Например, теперь доступны 64-разрядные стоечные серверы, и хотя консолидация нагрузки виртуализации является отличным примером использования оборудования такого масштаба, Берк считает, что сейчас мало задач, способных в полной мере занять 64-битные процессоры.

Увеличение эффективности самого оборудования за счет оптимизации работы, в сочетании с предоставлением администраторам таких преимуществ виртуализации, как гибкость и эффективность – ключевые позиции Red Hat Enterprise Linux 6. Усовершенствованный ввод-вывод – еще одна разработка в RHEL 6, связанная с виртуализацией. По словам Берка, виртуализация не является приемлемой платформой для IO-интенсивных рабочих задач, типа обмена сообщениями и баз данных, из-за накладных расходов до 40%, которые, тем не менее, в RHEL 6 снижаются до 5% благодаря сотрудничеству команды инженеров Red Hat и их коллег – создателей оборудования.

«Этот новый класс оборудования и оптимизация Red Hat Enterprise Linux позволяют виртуализации стать эффективным средством развертывания для всех категорий рабочих нагрузок. Это открывает двери гибкости, независимости от платформы и более широкому применению облаков», заявляет Берк.

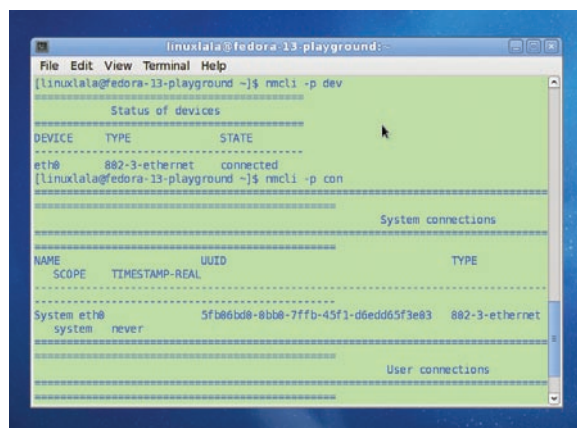
Хен — это клево

Red Hat Enterprise Linux 5, выпущенный в 2007 году, ввел *Xen* как предпочтительную платформу виртуализации. Тем не менее, с тех пор Fedora перешла на *KVM*, а *Xen*, хотя и будет поддерживаться еще несколько лет для клиентов RHEL 5, исчезнет из RHEL 6. Вместо этого пользователям будет предоставляться средства для преобразования виртуальных образов *Xen* в *KVM*, если они этого захотят.

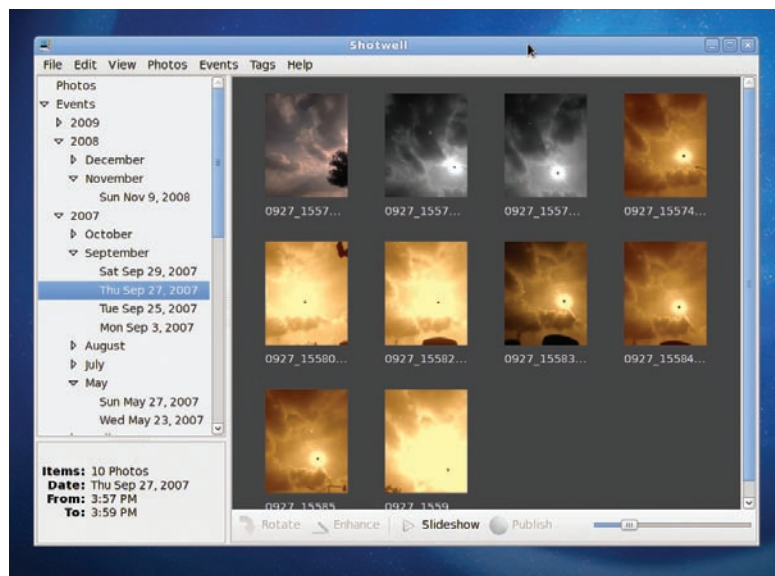
RHEL считают стандартом *де-факто* для Linux-сервера уровня предприятия, и мы попросили Берка указать на некоторые ключевые моменты развития Red Hat Enterprise Linux за последние годы. Обычно приобретаемая задорого у третьих лиц, виртуализация, предлагаемая в Red Hat Enterprise Linux 5, привнесла в дата-центры операционную эффективность по массовым расценкам.

Что не менее важно, поддержка безопасности через SELinux, изолирующая эксплойты системы, обеспечила устойчивость RHEL к атакам.

И, наконец, малая латентность в режиме реального времени. Плечом к плечу с сообществом, инженеры Red Hat разработали ряд мер, делающих временные рамки реагирования системы весьма предсказуемыми и управляемыми. Они дают более точный хронометраж и обеспечивают приоритетным приложениям больше ресурсов для выполнения. Выгоду от этого получают все виды



► **Nmcli**, консольный инструмент для контроля *Network Manager*, пока не догнал графического собрата.



► **Shotwell** – замена инструмента *F-Spot Photo Manager* в Fedora 13. Другое замененное Моно-приложение – *Tomboy*.

задач, от баз данных до отображения графики и высокоскоростного обмена сообщениями.

Ветки Fedora

Альтернативные версии Fedora, предлагающие индивидуальные наборы приложений и настройки для удовлетворения нишевой группы пользователей, называются «spins» – ответвления. Игровые, KDE и *Xfce*-ответвления существуют уже давно, а новое дополнение – Sugar-on-a-Stick (SoaS).

Изначально разработанную как часть проекта для школьников «Ноутбук – каждому ребенку [One Laptop Per Child, OLPC]», среду обучения Sugar теперь можно рассматривать как независимую от аппаратуры OLPC: она может работать с любым накопителем USB и занимает ничтожные 500 МБ в виде ISO-файла. Дети могут получить к ней доступ с помощью любого компьютера, дома или в школе, без привязки к ноутбуку.

Также довольно популярна ветка Security Spin, который поддерживается сообществом тестеров безопасности и разработчиков и предлагает безопасную тестовую среду для работы над аудитом безопасности, расследованиями и восстановлением системы.

Вы можете создать свое собственное ответвление на Live USB, с помощью легкодоступного инструмента:

```
su -c "yum install liveusb-creator"
```

Игра чисел

Вполне естественно желание сравнить Fedora с Ubuntu, но мы должны отметить, что оба эти дистрибутива имеют очень четкие истоки и цели и направлены на разных конечных пользователей. Тем не менее, в большинстве конкурсов на популярность Fedora в лучшем случае играет роль второй скрипки после Ubuntu. И это несмотря на инновации в Fedora и стойкий опыт производства дистрибутива, каждому хоть что-то да предлагающего.

Если вы пользуетесь неким дистрибутивом и вполне им довольны, это не повод поливать грязью другие дистрибутивы. И все же, чаще всего, единственным аргументом против Fedora является то, что он не похож на Ubuntu.

«Исторически, в Fedora всегда делали упор на эмпирические оценки и поощрения сообщества приверженцев. Проект Fedora продолжает, например, публиковать полную и достоверную статистику, касающуюся загрузки и пользовательских профилей оборудования», аргументирует Фрилдс. Страница статистики отслеживает эти цифры по адресу <http://fedoraproject.org/wiki/Statistics>. **LXF**

OSM: Сам себе картограф

Википедия постепенно становится именем нарицательным, но оказывается, аналогичный «колхозный» подход можно применить не только к текстам, но и к картам. Участник проекта OpenStreetMap **Сергей Семенов** делится опытом.



Наш эксперт

Сергей Семёнов

Системный администратор, увлекающийся велоспортом, альпинизмом, мото- и пешими путешествиями, которые бережно переносит в OpenStreetMap.

С древних времён люди рисовали карты, чтобы не блуждать бессмысленно по просторам нашей любимой планеты, но блуждать со смыслом. Полезность сего ясна не сразу, и я – как, наверное, многие – оставлял тематику картографии за пределами своих интересов. До тех пор, пока волею судьбы не пришлось мне заблудиться на Байкале. Накуролесив по местным просёлочным дорогам 250 км, по возвращении домой во Владивосток я стал подумывать о GPS.

Карты — какие они?

Что GPS без карты? Просто набор координат на чёрно-белом фоне. Погрузившись в Интернет, я вдруг понял: огромное количество объединений людей создаёт десятки разных карт одних и тех же участков местности – разной детализированности, наполненности, платности и правил распространения. В силу их закрытости новые карты не могут наследовать уже собранную информацию, и потому рисуются с нуля или почти с нуля, тем самым увеличивая большой список всё тех же неполных карт. А ведь нужна всего одна карта – хорошая. И чтобы обновлялась регулярно, и чтобы содержала максимум информации, и чтобы объединяла усилия всех картографов воедино и позволяла поправить/добавить нужный объект, если такой найдётся.

К своей радости, я нашёл такой проект – это OpenStreetMap (openstreetmap.org), объединивший огромное количество людей

над созданием открытой, бесплатной карты мира. Один из его главных плюсов – можно стать полноправным участником и всё-таки нанести на мировую карту свой дом, село, где родился, проход по ущелью гор и тот прекрасный родник, у которого напился сам и который, быть может, утолит жажду такому же скитальцу с GPS-навигатором.

И что с этим делать?

Итак, навигатор есть, карты есть. Осталось подружить их с Linux и друг с другом и как-то начать этим пользоваться. У навигаторов есть две основные функции. Первая и главная – выдавать информацию о текущем местоположении. И вторая, не менее важная – записывать «трек», то есть список из точек с координатами, высотой и временем, когда путник находился в этой точке. Этот трек и показывает дорогу, которой путник прошёл. Треки можно выгружать, редактировать, преобразовывать в пути, дороги, но это тема дальнейшего повествования, а пока нам нужно загрузить карты в наш навигатор.

В случае навигаторов Garmin, карта представляет собой один файл с именем `gmapsupp.img`, лежащий на флэш-карте навигатора в директории **Garmin** (именно с большой буквы). Итак, вынимаем карту, подключаем её к компьютеру через переходник как обычную «флэшку», создаём в корне директорию **Garmin** (если её там нет). Далее нам нужно скачать саму карту.

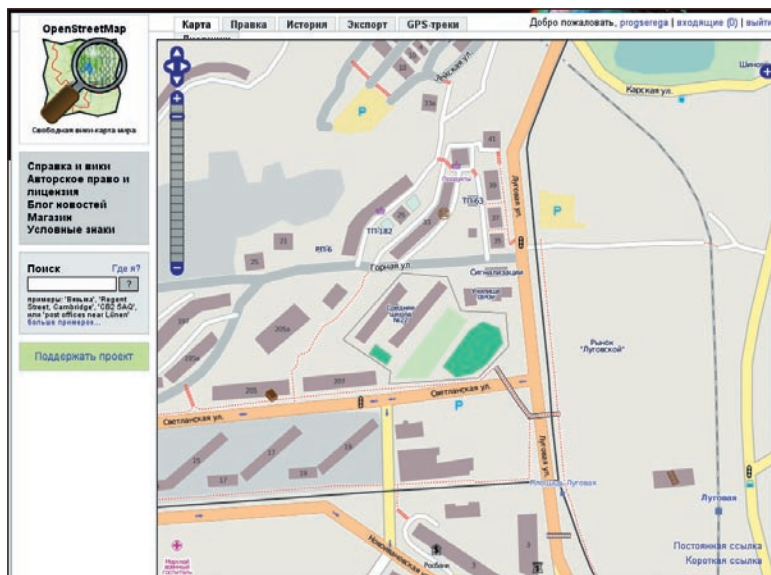
По умолчанию карты OpenStreetMap конвертируются в формат навигаторов Garmin и выкладываются на сайт <http://gis-lab.info/data/mp/> раз в сутки. Однако там публикуются архивы, содержащие набор файлов, которые нужно открывать с помо-

Какой же взять навигатор?

Когда-то я тоже задавал себе этот вопрос, но скоро догадался, что лучше не открывать Америку, а спросить у специалистов. Поэтому, доверившись открытости проекта OpenStreetMap, я пошёл прямоком на http://wiki.openstreetmap.org/wiki/RU:Main_Page и стал искать информацию о том, какой же навигатор будет полноценно работать с OSM и с Linux.

На странице http://wiki.openstreetmap.org/wiki/GPS_Reviews дано описание большого (даже слишком) количества устройств, но, как выяснилось, большинство «опытных OSM'щиков» предпочитает Garmin 60c(s)x из-за надёжности и точности позиционирования. Почитав http://wiki.openstreetmap.org/wiki/Garmin/GPS_series, я понял, что это оно, вот только... Только Garmin Dakota уж больно интересен. И с этим интересом я пошёл в магазин, где мы поставили эксперимент над Garmin 60csx и Garmin Dakota. Пока Dakota упорно искал спутники, 60-ка уже показывала координаты. Выбор был сделан.

➤ OpenStreetMap – открытая карта мира.



цию специального редактора карт. Мы же можем поступить проще – скачать уже полностью сформированную карту России для Garmin с торрент-трекера: <http://rutracker.org/forum/viewtopic.php?t=1911128> (обновляется примерно раз в месяц). Загрузив файл **gmapsupp.img**, скопируйте его на карту навигатора. Вставьте ее в устройство, включите, и... вуаля – у нас в навигаторе открытая карта всей России!

А как редактировать?

В редактировании, как и в заливке, есть несколько способов. Есть простые – и есть удобные. Рассмотрим самый простой – редактор *Potlatch*, написанный на Flash. Чтобы воспользоваться им, достаточно зарегистрироваться на сайте openstreetmap.org, войти под своим именем и нажать на кнопку Правка вверху экрана. Открывается *Potlatch*; выбрав в нем режим правки, можно приступить к рисованию, например, своего дома.

Ключевым элементом является точка. Точки формируют линии, линии образуют закольцованные полигоны (многоугольники). Вот в общем-то и вся сложность. Чтобы различать, какая линия есть река, а какая – дорога или тропинка, с каждым примитивом можно связывать тэги, у которых есть какое-либо значение. Как название тэга, так и его значение может быть любым. Однако для того, чтобы эти тэги влияли на отрисовку данного объекта на карте в браузере или в навигаторе, их следует брать из определённого списка тэгов, часто употребляемых сообществом. Его можно найти по ссылке http://wiki.openstreetmap.org/wiki/RU:Map_Features.

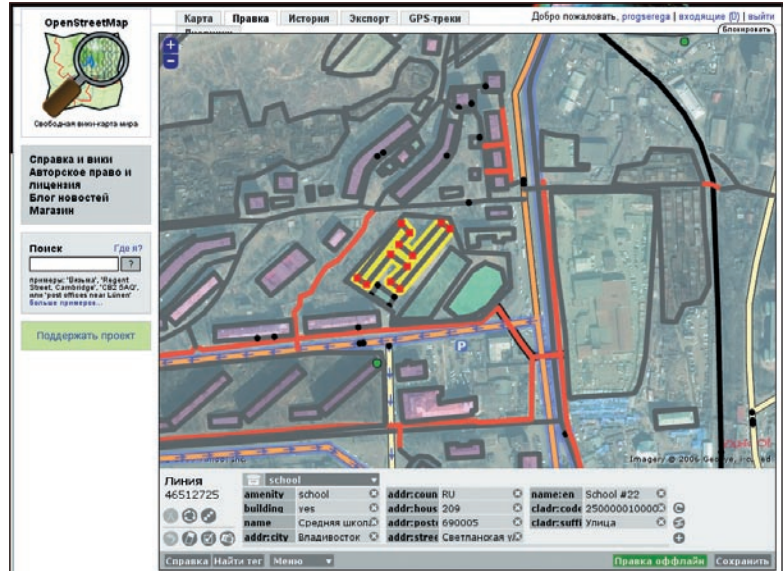
Координаты и наборы тэгов, переданные на сервер openstreetmap.org, впоследствии обрабатываются и либо конвертируются в файл карт для различных навигаторов, либо отрисовываются в виде изображений на карте, которую мы видим на главной странице проекта.

Закрепим изученное на практике. Например, мы хотим нарисовать наш дом; для этого на карте openstreetmap.org находим свой район проживания, жмём Правка, выбираем «Правка с сохранением» и рисуем прямоугольник, последняя точка которого соединяется с начальной. Теперь нужно сообщить системе, что это дом. Ниже есть кнопка со знаком +; подсветка скажет, что это значит «Добавить новый тэг». Нажмите кнопку и укажите вместо key название нашего тэга – `building`. Значение выставьте в «yes». Известные тэги появляются в выпадающем списке. Нажимаем Сохранить. Через некоторое время на основной карте мы сможем наблюдать наш дом. Так же рисуются дороги, перекрёстки, точки интереса и т.п. Более того, нажав кнопку Задание настроек, можно выбрать подложку в виде космических фотоснимков района от Yahoo! и прочих, что облегчит рисование объектов.

Однако не стоит забывать, что всё это имеет много меньшую точность, чем сами треки с навигатора. Космоснимки, к примеру, могут иметь смещение в 50 и более метров, то есть рисовать дорогу стоит не столько на основе спутниковых карт, сколько по трекам навигатора. Их можно отобразить, нажав кнопку Показать GPS-треки. Если другие «мапперы» загружали на сервер треки, проходящие по вашей местности, то они отобразятся. В режиме редактирования можно смотреть, какие тэги используют «мапперы», рисуя объекты, соседние с вашими, просто выделив соседний объект. Для всех этих задач есть соответствующее ПО. Давайте рассмотрим все процессы работы с GPS, используя инструментариум Linux.

Исходные данные

Базовым ПО для работы с навигатором Garmin в Windows является *MapSource* (http://www8.garmin.com/support/download_details.jsp?id=209). Его практически полным аналогом в Linux (и не только) служит *Qlandkarte GT* (www.qlandkarte.org). Одним из замеченных мной отличий является отсутствие поддержки поиска по ад-



ресам в населённых пунктах. Иными словами, файлы для сборки карт, полученные, например, с сайта <http://gis-lab.info/data/mp/>, с помощью *Qlandkarte GT* можно успешно загрузить в навигатор, однако поиск улиц в городах работать не будет. В остальном же программа справляется со своей задачей.

Дополнительно к *Qlandkarte GT* необходимо установить набор драйверов для различных устройств от Garmin. В моем дистрибутиве (Gentoo) это пакет *sci-geosciences/qlandkartegt-garmindev*. Дело в том, что драйвер для устройств Garmin в ядре (*garmin_gps*) не слишком хорошо работает, поэтому перед запуском *Qlandkarte GT* его нужно либо выгрузить, либо замаскировать соответствующим образом:

```
# cat /etc/modprobe.d/blacklistgrep garmin
# blacklist garmin_gps
```

Установив *qlandkartegt* и *qlandkartegt-garmindev*, подключим GPS-устройство к компьютеру через USB и запустим *Qlandkarte GT*. В настройках (Установки > Основные > GPS-устройство) выставьте кодировку в `sr1251` (в версиях старше 0.15.x такой возможности нет), укажите тип устройства и его последовательный порт – `/dev/ttyUSB0`. Далее, воспользуйтесь меню Трек > Загрузить и Путевые точки > Загрузить, чтобы скачать с устройства треки и точки интереса. Слева внизу выбираем вкладку Карты, два раза щелкаем по OSM – в результате подложка будет подгружаться с openstreetmap.org. Загруженные точки и треки можно выгрузить в виде GPX-файла для дальнейшей работы или публикации на openstreetmap.org. Это делается через меню Файл > Сохранить GEO-данные; нужно выбрать формат GPX.

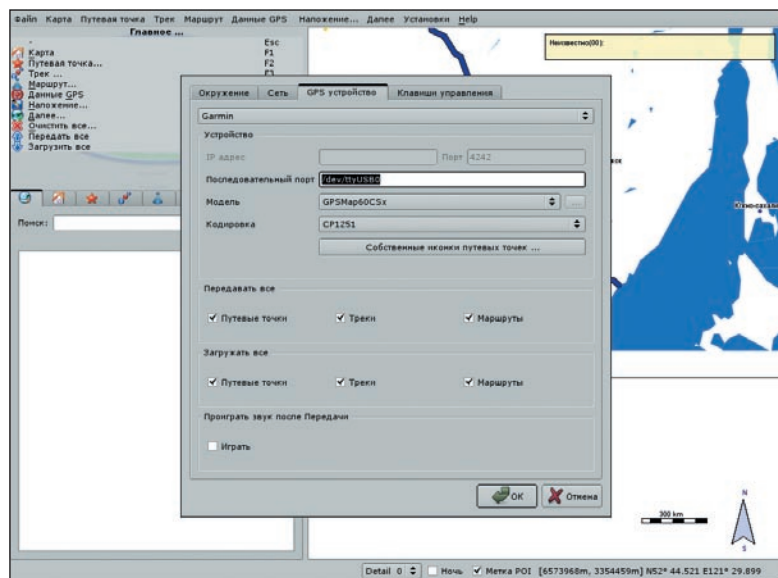
Данный способ хорош для эпизодического использования, но не очень удобен для регулярной работы с треками. Более гибким решением здесь будет воспользоваться утилитой *GPSSbabel* (www.gpsbabel.org), которая должна быть в репозиториях вашего дистрибутива (в моем она называется *sci-geosciences/gpsbabel*). Утилита позволяет скачивать и при надобности обрабатывать треки и точки, а также конвертировать их в различные форматы.

```
# gpsbabel -w -r -t -i garmin,encoding=windows-1251 -f usb: \
-x track,pack \
-x discard,hdop=5,vdop=5 \
-x position,distance=5m \
-x interpolate,time=60 \
-o gpx -F out_file.gpx
```

Что здесь происходит?

- » **-w** – скачиваем и точки интереса в том числе
- » **-r** – скачиваем маршруты.
- » **-t** – скачиваем треки.

» Редактор *Potlatch* позволяет вносить изменения в карты, не покидая сайта OpenStreetMap.



➤ Настроить соединение с GPS-навигатором в Qlandkarte GT достаточно просто.

- **-x track.pack** – этот фильтр объединяет треки отдельных переходов в один большой.
- **-x discard,hdop=5,vdop=5** – этот фильтр выбрасывает из трека точки, для которых горизонтальная или вертикальная мера ошибки больше 5 (как HDOP, так и VDOP связаны с метрами). На практике этот фильтр действительно эффективно отсеивает точки, внезапно «высочившие» из траектории. Значение «5» – эмпирическое; иногда я указываю даже «10».
- **-x position,distance=5m** – объединяет точки, расстояние между которыми меньше 5 м (короткие остановки).
- **-x interpolate,time=60** – интерполирует трек так, чтобы между соседними точками было не более 60 секунд (это один из самых полезных фильтров для создания геотэгов!).

Геотэги мы рассмотрим несколько позже. Смысл всех остальных опций ясен из описания. Используя *GPSBabel*, можно написать скрипт, который при вызове будет скачивать треки, править их и складывать в нужное место с именем, состоящим из даты, а затем формировать символическую ссылку на него вида «новые_треки.gpx». Скрипт можно вызывать через *udev*, при подключении определённого типа устройства (вашего GPS-навигатора) и тому подобное – автоматизация рутины ограничивается лишь вашей фантазией.

Здесь стоит упомянуть об одной досадной неприятности. В базовой инсталляции *GPSBabel* некорректно обрабатывает параметры кодировки, поэтому русские имена в точках интереса будут в «кракозябрах». Во всяком случае, так ещё было в версии 1.3.6. Чтобы поправить это недоразумение, есть патч от Александра Дежина (<http://forum.openstreetmap.org/viewtopic.php?id=3753>). Вы также найдёте его на **LXFDVD**. Это исправление было отправлено разработчикам – возможно, вскоре оно появится в основной ветке.

Скачав трек, можно выложить его на сайт: openstreetmap.org > GPS-треки > Показать только ваши треки, или Передать новый трек на сервер > Обзор. По умолчанию я выставляю тип Общедоступный. Какой выставите вы – решать вам. Общедоступным треком смогут воспользоваться другие «мапперы», редактирующие тот же район карты.

Создаем карту

Пришло время заняться полноценным редактированием карты с помощью соответствующего ПО. На выбор есть два основ-

ных редактора: *JOSM* (на java) и *Merkaartor* (на Qt). Я предпочитаю *Merkaartor* (www.merkaartor.org). Установите его, как обычно (пакет *sci-geosciences/merkaartor*) и запустите командой `merkaartor`. Далее, зайдите в Tools > Параметры > Данные и укажите ваши имя и пароль на openstreetmap.org, чтобы изменения выгружались под вашей учётной записью.

Загрузите трек: Файл > Открыть, и укажите наш GPX-файл с треками и точками интереса. Нажмите кнопку Загрузить ещё, чтобы подгрузить карту OSM в видимую на экране область. Теперь можно приступать к редактированию. У каждого примитива есть всё те же тэги: манипулируя ими, а также не забывая сверяться с треками и «подглядывая», как редактировали другие «мапперы», можно быстро понять логику. После завершения редактирования нужно выгрузить изменения на openstreetmap.org, нажав кнопку Опубликовать.

Однако здесь есть некоторые тонкости – это слои и подложки. Треки, точки, подгружаемая информация, редактируемые элементы – всё это находится на своём слое. Выгружается на сервер только «Слой черновика». Соответственно, чтобы сформировать из трека дорогу, нужно нажать правой кнопкой по слою «ACTIVE LOG #2» (в моём случае, см. рисунок) и выбрать «Извлечь редактируемый слой». Уберите видимость со слоя трека (чтобы не мешался при редактировании) – и теперь у нас есть полноценный путь. Так как зачастую нужен не весь путь, а лишь часть его, то эту часть нужно отделить, выделив один из узлов и нажав Дорога > Разделить. «Откусив» такими операциями кусочек трека, можем навешивать на него тэги. Тут-то и проявляется тонкость работы со слоями: мы редактировали слой «Извлечение ACTIVE LOG #2», а чтобы перенести то, что мы там наредктировали, на «Слой черновика», нужно выделить наш новый пешеходный участок, нажать Правка/Скопировать и вставить фигуры через Правка > Paste Figure(s). После этого путь появится на слое черновика. Проверить это можно, включив/выключив видимость соответствующих слоёв. И ещё раз напомним, что загружая карту по кнопке Загрузить, не забывайте выставлять галочку «Также загрузить необработанные треки GPS», чтобы подгрузить треки других «мапперов»

«Изменения появляются на карте спустя некоторое время.»

в этом районе, увеличив этим точность ваших объектов. К слову, на текущий вид карты можно добавить закладки через меню Вид > Закладки > Добавить.

Чтобы можно было читать подписи к точкам интереса, нужно убрать галочку «Только чтение» со слоя «Маршрутные точки». Для удобства редактирования можно подключать слой подложки – например, спутниковые снимки Yahoo! или Космоснимки, если покрытие в вашем городе у них лучше. Для выбора подложки в слоях нажмите правой кнопкой по слою Карта и отметьте «Yahoo – WMS». При желании можно настроить локальный кэширующий сервер для подложки, чтобы при каждом изменении масштаба не посылался запрос к Yahoo. Сам *Merkaartor* не кэширует карты из-за запрета этого в лицензии к картам.

Опубликованные изменения появляются на карте спустя некоторое время: от двух минут до суток. Для удобства каждые сутки делается выгрузка и конвертация карт по краям и областям страны: <http://gis-lab.info/data/mp/>. Скачав этот архив, распакуйте его и откройте в *Qlandkarte GT* (Файл > Загрузить карту) и выберите файл `osm.tdb`. Загрузив одну карту, вы можете захотеть совместить её с другой – например, картой рельефа России: <http://rutracker.org/forum/viewtopic.php?t=1686360>. Загрузите её в *Qlandkarte GT* аналогичным образом. На вкладке карт выберите нужную, жмём F5 и нажимаем на область карты. В данном случае я выбрал Приморский край. Выбираем карту рельефа, жмём F5, нажимаем левую кнопку мыши и выделяем участок блоков,

карты высот которых мы хотим выгрузить для навигатора. К слову, рельеф всей России выгрузить не удастся, из-за ограничений платформы Garmin. В левом нижнем углу выбираем Выбранные карты и жмём или Карта > Передать, чтобы передать её на подключённый к USB GPS-навигатор, или Экспортировать карту, чтобы сохранить её в файл **gmapsupp.img** на компьютере для последующего копирования на карту навигатора.

Приятные мелочи

На практике обычно выясняется, что большое количество информации требуется заносить куда-то помимо навигатора. Сюда попадают номера домов, названия кафе, кинотеатров, памятников, железнодорожных станций и прочее. Можно использовать блокнот, но удобнее – обычный цифровой фотоаппарат. Логика состоит в следующем: вы синхронизируете время на обоих устройствах и, рисуя трек, фотографируете любую дополнительную информацию. В EXIF-данные к каждому снимку фотоаппарат запишет время создания, сопоставив которое в дальнейшем с аналогичным временем точки трека, мы получим GPS-координату фотографии, которую специальной утилитой можно дописать к каждому снимку. После этого фотографии можно загружать в *Merkaartor* и, видя их на треке, чётко понимать, что же там действительно находится в реальности. В этом нам поможет утилита *Happy Camel* (happycamel.sourceforge.net).

Как всегда, есть несколько вариантов использования. Первый – каждый раз перед «маппингом» выставлять время фотоаппарата по времени навигатора. Тогда по приходу домой вы скачиваете с фотоаппарата снимки, трек с навигатора, заходите в директорию с фотографиями и запускаете команду:

```
#/usr/bin/happycamel -x y --utc-offset=+10:00
--exiftool=y --elevation=2
--use-outside=y -t трек.gpx `ls *.jpg`
```

где **--utc-offset=+10:00** – сдвиг по времени относительно Гринвича (во Владивостоке – 10 часов).

Если всё пройдёт хорошо, то в EXIF-блок каждой фотографии будет дописана информация о GPS-координатах, где была проведена съёмка. Однако это слишком сложно с точки зрения количества действий. Здесь есть простор для автоматизации. Второй вариант – каждый раз перед «маппингом» вы фотографируете экран навигатора, на котором должны быть видны текущая дата и время, и, не корректируя время в фотоаппарате, продолжаете маппинг. Придя домой, вы запускаете в директории с фотографиями следующую команду:

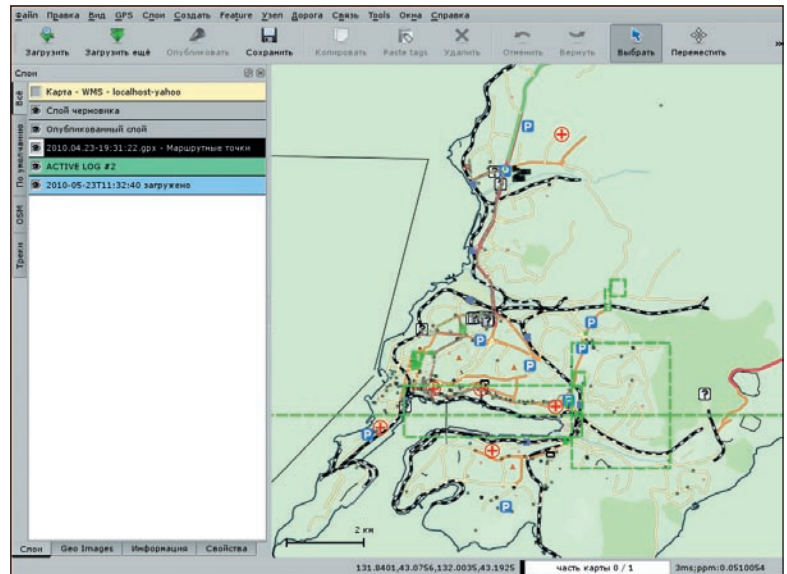
```
#/usr/bin/happycamel -x y --calibrate=файл_фотографии_времени_навигатора.jpg \
2010-02-18-09-51-19 --fix-dates=y --utc-offset=+10:00
--exiftool=y \
--elevation=2 --use-outside=y -t трек.gpx `ls *.jpg`
```

где

» файл_фотографии_времени_навигатора.jpg – фотография навигатора

Полезные ссылки

- » Коды КЛАДР: <http://osm.tesoro-shop.ru/>
- » Форум русского сообщества OSM: <http://forum.openstreetmap.org/viewforum.php?id=21>
- » Вики OSM: http://wiki.openstreetmap.org/wiki/RU:Main_Page
- » Отображение маршрутов транспортных средств: <http://latlon.org/pt>
- » Видео «Год редактирования OpenStreetMap»: <http://www.youtube.com/watch?v=hw1qUremx4Y>
- » Прокладка маршрута по OSM: OpenRouteService.org



- » 2010-02-18-09-51-19 – время на навигаторе, запечатлённое на этой фотографии, в формате год-месяц-день-час-минуты-секунды
- » трек.gpx – трек с навигатора

» Подобно *GIMP*, редактор карт *Merkaartor* поддерживает работу со слоями.

В результате скрипт сам высчитает смещение времени между фотоаппаратом и навигатором с точностью до секунды и поправит время фотографий, а также допишет координаты в соответствии с новым временем.

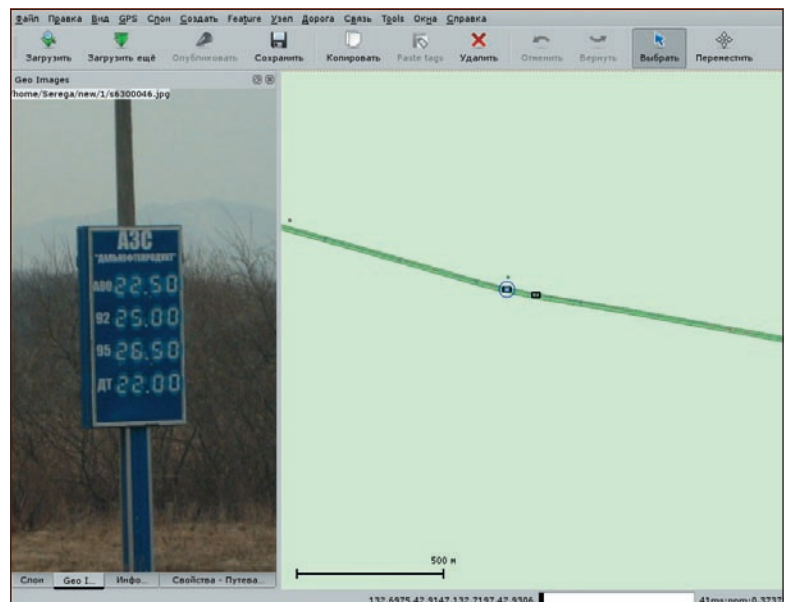
Далее открываем *Merkaartor*, загружаем туда трек и фотографии (Файл > Открыть, выделить все снимки). *Merkaartor* спросит, какой слой использовать – выберите слой трека «ACTIVE LOG #2» (в моём случае) и согласитесь снять с него флаг «только чтение». В результате на треке мы увидим чёрные квадраты; выбрав какой-либо квадрат мышью, можно посмотреть изображение, не забыв перед этим включить соответствующее окно через меню Окна > Панели > Geolmage.

К слову, *Merkaartor 0.15* некорректно воспринимает пути с русскими символами, поэтому для загрузки фотографий *Merkaartor*ом мне пришлось делать ссылку на фотографии на английском языке:

```
# ln -sf -/Фотографии/Новое/2010.04.11-23:56:35 new
```

Приятного «маппинга!» **LXF**

» Геотэггинг позволяет привязывать цифровые снимки к точкам на местности.



Программируем периферию

Принято говорить, что Linux умеет все, кроме варки кофе – но это неправда. **Андрей Боровский** не будет пытаться воспроизвести старое HOWTO на эту тему, но представит вам информацию, по которой вы сможете написать его сами.



Наш эксперт

Андрей Боровский теперь может включить свет на кухне, не вставая из-за компьютера.

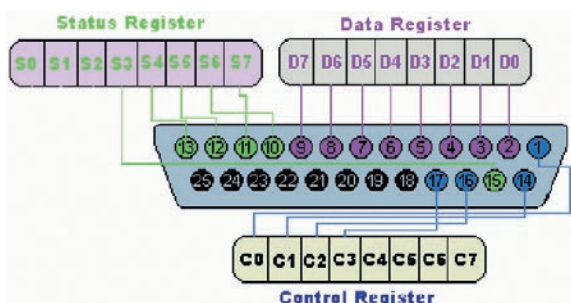
Лет 50 назад в моде были прогнозы относительно того, как будет выглядеть повседневная жизнь в XXI веке. Читать их в наше время и забавно, и интересно, и поучительно одновременно. Сами же футурологи прошлого, глядя на наше время, неизбежно испытали бы досаду. И дело не только в том, что мы до сих пор не живем до двухсот лет и не летаем на Луну по путевкам. Посмотрите хотя бы на то, как мы используем компьютеры! Вместо того, чтобы сочинять музыку и прокладывать маршруты для звездолетов, мощнейшие вычислительные средства используются для обмена колкостями в Живом журнале и просмотра видео сомнительного содержания. В этой статье я предлагаю другое применение этих мощностей – в качестве электронного мозга для управления устройствами нашего дома. Хорошему мозгу нужны хорошие мускулы (или хотя бы периферические нервные окончания), так что мы сосредоточимся на проблемах сопряжения домашнего компьютера с внешними устройствами.

Устройства, которые мы рассмотрим, отличаются простотой. Детали, необходимые для их сборки, найдутся в арсенале любого заядлого компьютерщика. Если вы не любите паять – не беда, обойдемся и без пайки. Такой подход имеет обратную сторону: используемые нами решения не являются профессиональными и, скорее всего, не подойдут для устройств, имеющих хотя бы отдаленное промышленное назначение. Строго говоря, это именно то, что называется словом «hак». Используемые мной подходы не представляют ничего существенно нового. Описания многих из этих трюков можно найти в Интернете, где они ориентированы в основном на ОС Windows. Но когда я занялся интеллектуализацией своего дома, то обнаружил, что Linux гораздо лучше подходит для решения этих задач, нежели ОС от Microsoft. Об этом я и собираюсь вам рассказать. Прежде чем продолжить, произнесу обязательное ритуальное заклинание: следуя приведенным здесь советам, вы действуете на свой страх и риск. Я заранее снимаю с себя ответственность. Все приведенные электрические

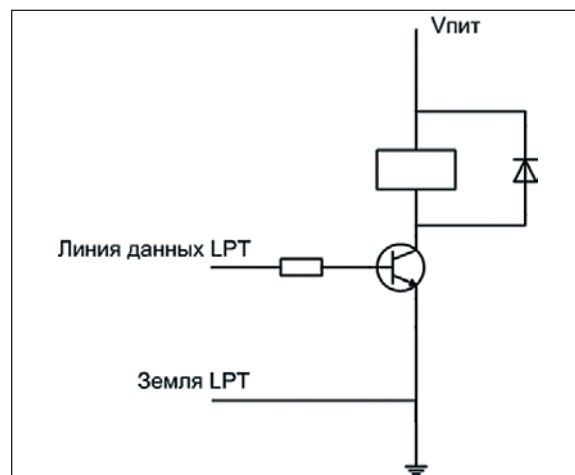
схемы носят приблизительно-показательный характер (параметры резисторов и конденсаторов не указываю). Если вы задумаете реализовать их – уточните параметры самостоятельно.

Наш лучший друг – параллельный порт

Параллельный порт как нельзя лучше подходит для наших целей. Стандартные напряжения сигналов на его выводах соответствуют уровням напряжения TTL-электроники, и мы обладаем над ними полным контролем: можем устанавливать и сбрасывать их на сколь угодно продолжительное время. Хотя изначально параллельный порт предназначался для вывода данных на принтер, в более поздних версиях стандарта были добавлены возможности чтения данных с внешнего устройства (те из вас, кто сел за компьютер в 90-х или ранее, наверняка помнят сканеры, которые подключались к компьютеру через параллельный порт). Впрочем, даже самый «тупой» параллельный порт умеет считывать внешние сигналы – например, сигнал о том, что в принтере закончилась бумага. Схема контактов параллельного порта представлена на рис. 1. Мы располагаем восемью линиями для передачи данных (в базовой версии порта – от компьютера ко внешнему устройству), тремя линиями управления (также работающими на вывод) и пятью линиями состояния (ввод данных). На каждой из этих линий может быть установлена логическая единица (+5 В) или ноль (0 В). Для управления внешним устройством – например, реле – мы выводим уровень логической 1 на соответствующие линии данных порта (рис. 2).



➤ Рис. 1. Так выглядит параллельный порт – точнее, его схема.



➤ Рис. 2. Подключение реле к параллельному порту.

Управлять параллельным портом из Linux можно двумя способами: с помощью файла устройства `/dev/parport*` и путем непосредственного доступа к адресам порта. В обоих случаях для получения доступа к порту программа должна обладать правами `root`.

Рассмотрим, как решить эту задачу методом непосредственного доступа к портам (здесь нужно различать «порт» в том смысле, о котором мы говорили выше, и «порт устройства» как канал обмена данными, отображенный в адресное пространство).

Программа, представленная на листинге, предлагает пользователю ввести число и устанавливает сигналы на линиях данных параллельного порта в соответствии с маской его битов.

```
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/io.h>
#include <iostream>
using namespace std;
int main()
{
    int addr = 0x378;
    if (ioperm(addr,1,1) {
        cout << "Невозможно установить разрешения
на доступ к порту. Вы должны быть root'ом." << endl;
        return -1;
    }
    char i = 1;
    while (i) {
        cout << "Введите число";
        cin >> i;
        outb(i, addr);
        cout << "Записано" << endl;
    }
    return 0;
}
```

0x378 – адрес регистра данных первого параллельного порта компьютера. Если в системе есть второй параллельный порт, доступ к его регистру данных осуществляется через ячейку по адресу 0x278. Чтобы получить доступ к адресу регистра порта, программа должна запросить и получить соответствующие полномочия. Мы делаем это с помощью функции `ioperm()`, которая доступна только процессам, выполняющимся с правами `root`. Первый ее аргумент – начальный адрес порта устройства, к которому мы хотим получить доступ. Второй аргумент – размер области памяти в байтах. Третий аргумент указывает, должен ли доступ быть включен (1) или выключен (0). Если функция выполнена успешно, она возвращает ноль, в противном случае – -1. Получив доступ к порту устройства, мы можем записывать данные с помощью функций `outb()`, `outw()`, `outl()`, где `b` означает `byte`, `w` – `word`, `l` – `long`. Так, например, если мы запишем в регистр по адресу 0x378 число 4, на выводе 4-го разъема параллельного порта (линия D2) будет установлено напряжение, соответствующее логической 1. Проверить же работу программы, управляющей портом, очень просто. Для этого нам понадобятся несколько светодиодов и кабель Centronics, которым в былые времена соединяли компьютеры и принтеры (рис. 3). Поскольку параллельный порт предоставляет нам 8 линий данных, с помощью одного только регистра данных мы можем управлять восемью устройствами. Этим возможности параллельного порта не исчерпываются.

Как уже отмечалось, через регистр состояния мы можем передавать данные от внешнего устройства к компьютеру. Простейшее такое устройство – датчик размыкания. Он может, например, сообщить системе, что кто-то открыл входную дверь, окно или дверцу сейфа. Каким образом взаимодействует внешнее устройство и линия состояния порта? По умолчанию на линиях S3–S7



► Рис. 3. Кабель Centronics и светодиоды: созданы друг для друга!

установлены уровни логической 1, и маска регистра состояния выглядит как 11111000b. Чтобы установить на одной из линий состояние логической 1, мы просто замыкаем эту линию с землей порта через сопротивление 300–500 Ом. В результате значение соответствующего бита в регистре становится равным нулю. Таким образом, сброшенный бит свидетельствует о том, что цепь замкнута; установленный – о том, что разомкнута. Ниже приводится фрагмент программы, которая контролирует состояние датчика размыкания (предполагается, что датчик подключен к линии S5 (вывод 12)).

```
int main()
{
    int addr = 0x379;
    if (ioperm(addr,1,1) {
        cout << "Невозможно установить разрешения
на доступ к порту. Вы должны быть root'ом." << endl;
        return -1;
    }
    char i;
    while (1) {
        i = inb(addr);
        if (i & 32)
            cout << "Разомкнуто" << endl;
        else
            cout << "Замкнуто" << endl;
        sleep(1);
    }
    return 0;
}
```

Регистр состояния порта LPT1 доступен по адресу 0x379, а порта LPT2 – соответственно 0x279.

Как было отмечено выше, с параллельным портом можно работать и через файл устройства `/dev/parport*`, где * – число, обозначающее номер порта). После того как мы открыли устройство порта

```
fd = open("/dev/parport0", O_RDWR|O_NONBLOCK);
мы можем настроить его с помощью вызовов ioctl(). Например, команда PPCLAIM открывает программе доступ к порту:
if (ioctl(fd,PPCLAIM) <0)
    printf("Ошибка!");
```

Порт, занятый с помощью вызова `PPCLAIM`, может быть освобожден с помощью вызова `PPRELEASE`. Константы и макросы, имеющие отношение к параллельному порту, определены в заголовочных файлах `<linux/parport.h>` и `<linux/ppdev.h>`.

Для записи данных в порт его можно перевести в режим байтовой передачи:

```
int mode = IEEE1284_MODE_BYTE; /* or IEEE1284_MODE_NIBBLE, etc. */
ioctl(fd,PPNEGOT,&mode);
```

»

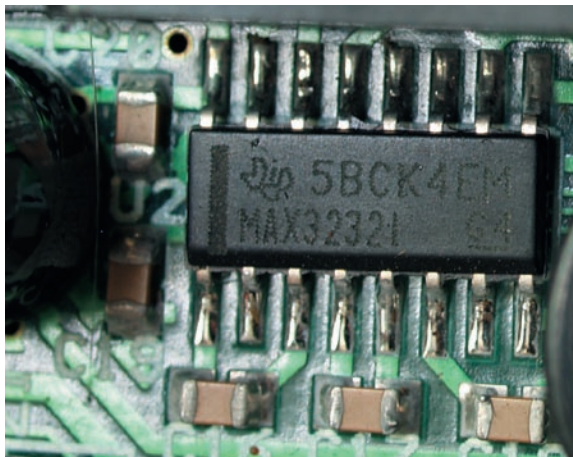
после чего можно использовать обычную функцию write(). Для чтения состояния порта мы воспользуемся вызовом PPRSTATUS:

```
int status;
ioctl(fd, PPRSTATUS, &status);
```

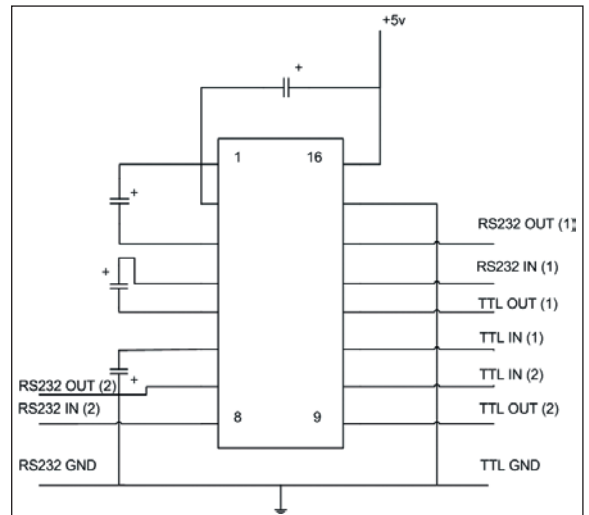
Последовательный порт

Этот порт гораздо меньше подходит для наших экспериментов – прежде всего потому, что он более «интеллектуален», нежели параллельный. Если в параллельном порту для передачи одного байта соответствующие уровни выставляются одновременно на восьми линиях, то в последовательном одна линия используется для передачи всех восьми битов (электроника, управляющая портом, «знает», когда заканчивается передача одного бита и начинается передача следующего). Использовать последовательный порт для полноценного управления внешним устройством можно только в том случае, если внешнее устройство оснащено такой же умной электроникой. Тем не менее, как минимум 4 линии последовательного порта можно использовать так же, как мы используем линии параллельного. Из этих четырех линий две предназначены для передачи данных от устройства сопряжения в компьютер, а две – наоборот, от компьютера к устройству сопряжения; их мы и рассмотрим. Речь идет о линиях DTR (Data Terminal Ready – компьютер готов к обмену данными) и RTS (Request To Send) – запрос на передачу данных. В отличие от самих линий передачи данных, сигналы на этих служебных линиях могут устанавливаться на сколь угодно длительное время (так же, как и в случае с параллельным портом). Если мы хотим передавать сигналы от внешнего устройства в компьютер, мы можем воспользоваться линиями DSR (Data Set Ready – внешнее устройство готово к работе) и CTS (Clear To Send – внешнее устройство готово к передаче данных).

Однако радоваться рано. Еще одна проблема связана с тем, что уровни напряжения, используемые последовательным портом для отображения нулей и единиц, отличаются от уровней, которые использует параллельный порт (и другая распространенная цифровая электроника). При работе с последовательным портом логической единице соответствует напряжение от -3 до -25 В, а логическому нулю – от +3 до +25 В (в других источниках – от -3 до -12 В и от +3 до +12 В соответственно). В моей системе напряжения составляли -12 В и +3 В. Их, мягко говоря, неудобно использовать напрямую. Скорее всего, вам придется выполнить преобразование уровней напряжения последовательного порта к стандартным уровням TTL. Самый популярный преобразователь такого рода – микросхема MAX232 (рис. 4) и ее многочисленные аналоги. Подробные сведения об этой микросхеме можно получить по адресу <http://www.maxim-ic.com/datasheet/index.mvp/id/1798>. Поскольку сигналы последовательного порта практически всегда приходится преобразовывать в сигналы логики TTL, подобные преобразо-



➤ Рис. 4. Микросхема-аналог MAX232 в естественной среде обитания.



➤ Рис. 5. Преобразователь на основе MAX232.

ватели есть почти в каждом устройстве, которое может получать данные с последовательного порта. Микросхема позволяет преобразовывать сигналы с четырех линий (две линии в прямом, две линии в обратном направлении). Общая схема с использованием преобразователя MAX232 приведена на рис. 5. Сигнал, поданный с последовательного порта на вход RS232 In (1), появится в преобразованном виде на выходе TTL OUT (1) и так далее.

Для программного управления последовательным портом мы воспользуемся штатными средствами Linux. Необходимые константы и макросы определены в файле <termios.h>. Почему termios? Дело в том, что в стародавние времена к последовательному порту Unix чаще всего подключали терминал (либо напрямую, либо через «прослойку» из двух модемов и телефонной линии). Ниже приводится простенькая программка, которая устанавливает и сбрасывает сигнал RTS с интервалом примерно в одну секунду.

```
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <stdio.h>
#include <sys/ioctl.h>
#include <termios.h>
int main() {
    int fd;
    int status;
    fd = open("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY);
    if (fd == -1) {
        perror("Не могу открыть файл устройства - ");
        return -1;
    }
    else
        fcntl(fd, F_SETFL, 0);
    while (1) {
        ioctl(fd, TIOCMGET, &status);
        if (!status) {
            printf("Порт недоступен\n");
            return -1;
        }
        status &= ~TIOCM_RTS;
        ioctl(fd, TIOCMSET, &status);
        sleep(1);
        ioctl(fd, TIOCMGET, &status);
        status |= TIOCM_RTS;
        ioctl(fd, TIOCMSET, &status);
    }
}
```

```
sleep(1);
}
return 0;
}
```

Последовательные порты представлены в Linux устройствами `/dev/ttyS*`, где * – число (как правило, от 0 до 7), соответствующее номеру порта. Порту COM1 соответствует устройство `/dev/ttyS0` и так далее. Префикс `tty` тоже намекает на то, что когда-то порты использовались для подключения терминалов. С этим же связана необходимость использовать флаг `O_NOCTTY` при открытии файла устройства – без него открытое устройство-терминал стало бы управляющим терминалом нашей программы (и весь стандартный ввод/вывод выполнялся бы через него). Флаг `O_NDELAY` говорит системе, что на наши операции ввода/вывода не должно влиять состояние сигнала DCD (обнаружение несущей данных). Получив дескриптор файла порта, мы можем управлять им с помощью `ioctl()`. Вызов `ioctl()` с константой `TIOCMGET` возвращает в третьем параметре маску битов состояния порта. Среди этих битов есть и интересующие нас DTR, RTS, DSR и DTS. Выделить соответствующие биты в маске состояния можно с помощью констант `TIOCM_DTR`, `TIOCM_RTS`, `TIOCM_DSR` и `TIOCM_CTS`.

Чтобы изменить состояние порта со стороны компьютера, нужно записать новую маску сигналов с помощью вызова `ioctl()` с командой `TIOCMSET`.

Программа управляет сигналами порта COM1. Если указанный порт в системе отсутствует, в маске битов состояния после вызова `ioctl()` со вторым параметром `TIOCMGET` будет записано значение 0.

Программу следует запускать от имени пользователя `root`, иначе ей будет отказано в доступе к файлу порта. Самый простой способ протестировать работу программы – проверить напряжения на контактах разъема. Напряжение замеряется между выводом 5 (земля) и 4 (DTR) или 7 (RTS). Для удобства можно подключить нуль-модемный кабель. Учтите только, что кабели бывают полные и неполные. В неполном кабеле интересующие нас линии DTR и RTS замкнуты на линии DSR и CTS того же устройства, то есть сопряженное устройство просто не получит сигналы через такой кабель. Помните также, что сигнал, который появляется на выходе DTR порта компьютера, будет перекинут кабелем на линию DSR внешнего устройства, а сигнал RTS появится, соответственно, на линии CTS. В кабелях, предназначенных для подключения модемов, линии не перекрещиваются.

Палочка радости

Еще один интересный для нас порт – порт джойстика или игровой. Скорее всего, он не выведен на заднюю панель вашего ПК, но его можно найти на материнской плате (рис. 6) или на звуковой карте (для экспериментов лучше всего подойдет старая ненужная «звуковуха»). В отличие от других рассмотренных портов, порт джойстика работает только на ввод данных. Он доступен по адресу `0x201`, и с ним можно работать с помощью функций `inb()` и `outb()` так же, как мы работаем с параллельным портом.

Старшие 4 бита байта считанного из регистра `0x201` отражают состояние кнопок джойстика. Их можно использовать так же, как мы использовали линии состояния параллельного порта. Младшие 4 бита интереснее: они связаны с четырьмя измерителями сопротивления (в интервале примерно от 0 до 100 кОм). В принципе, порт джойстика можно использовать как простой АЦП. Величина сопротивления измеряется по скорости зарядки конденсатора следующим образом: по умолчанию конденсаторы заряжены, и значения соответствующих им битов в регистре порта равны 1. В ответ на запись любого значения в порт конденсаторы разря-

Внешний COM-порт

Тот факт, что в наших программах взаимодействия с последовательным портом мы не обращаемся к регистрам оборудования напрямую, приносит нам неожиданную пользу. Благодаря этому мы можем использовать наши программы с такими устройствами, как переходник USB-Serial (см. рис.).

Если в Windows для подключения такого устройства придется специально устанавливать драйвера, то в Linux с ядрами 2.6.x все драйвера уже установлены и загружаются при подключении устройства автоматически. Этим устройствам соответствуют файлы `/dev/ttyUSB*`, где * – номер порта. Для работы с этими портами можно использовать те же вызовы `ioctl()`, что и для встроенных портов, но нужно учитывать некоторые нюансы. Если устройство по какой-то причине отключилось (а в ходе экспериментов с портом это происходит неоднократно), драйвер откроет новое, используя первое свободное имя файла, отличное от ранее использованного (в отличие от устройств `/dev/ttyS*`, для устройств `/dev/ttyUSB*` существуют только файлы, соответствующие реально подключенным устройствам). В моих экспериментах имена устройств

«прыгали» между `/dev/ttyUSB0` и `/dev/ttyUSB1`. В самом простом случае можно просто перебирать допустимые имена устройств и пытаться открыть их. Можно воспользоваться также содержимым директорий `/dev/serial/by-id/` и `/dev/serial/by-path/`. Эти директории содержат символические ссылки на порты, подключенные к системе в данный момент. Имена файлов-ссылок также более информативны, нежели простые имена файлов устройств.

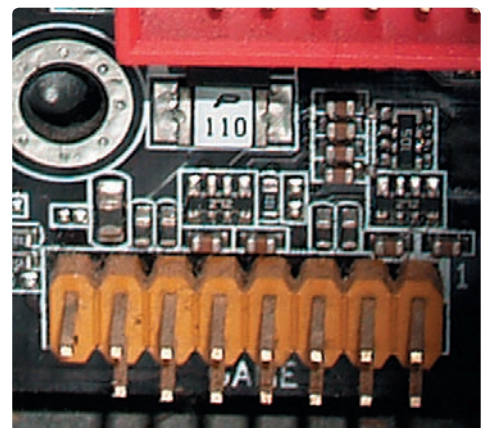


► Переходник USB-Serial (у нас – 25-контактный) превратит нетбук с Linux в центр управления оборудованием.

жуются, а соответствующие им биты принимают значение 0. Через некоторое время (пропорциональное величине измеряемых сопротивлений) конденсаторы снова зарядятся, и соответствующие биты регистра получат значение 1. Из сказанного очевидно, что для корректного измерения величины сопротивления необходимо точно измерить время с момента, когда биты состояния конденсаторов были сброшены и до момента их восстановления (интервал времени составляет несколько микросекунд). В многозадачной системе это, мягко говоря, непросто сделать. Непросто, но не невозможно!

Для обработки сигналов с порта джойстика мы можем воспользоваться «самым мощным оружием» пользовательских программ Linux – системным вызовом `ioctl()`. Этот вызов (который работает только для процессоров, основанных на архитектуре i386) позволяет программе, выполняющей с правами `root`, получить особые привилегии при работе с процессором и портами ввода-вывода. После вызова `ioctl()` с аргументом 3 программа получает возможность управлять флагом прерываний с помощью команд процессора CLI и STI. Заблокировав обработку прерываний (на как можно меньшее время) программа сможет изменить продолжительность времени зарядки с максимальной доступной точностью. Разумеется, с точки зрения «большого стиля» программирования, блокирование прерываний пользовательской программой не одобряется, так как приводит к снижению производительности системы (и это еще не самое худшее, что может быть), но мы еще в начале договорились, что будем играть не по правилам. LXF

► Рис. 6. Игровой порт ПК – спрятался под капотом.



» Через месяц Мы рассмотрим работу аналоговичных устройств по соединению USB.

Vala и Genie: Амбициозные новички



Уважаете C# как язык программирования, но не доверяете технологиям Microsoft? У Семёна Есилевского есть для вас кое-что.

При написании графических приложений от языка программирования требуется наличие многих понятий и конструкций, которые редко используются в консольных программах. К ним относятся, например, сигналы, лямбда-функции и возможность узнавать тип объекта во время выполнения программы. В той или иной степени эти возможности реализованы только в скриптовых языках, таких как Python или Ruby, а также в языках, использующих виртуальные машины — C# и Java. Однако использование скриптовых языков оправдано далеко не всегда, из-за их невысокой производительности. Язык Java явно не прижился в настольных приложениях для Linux, а «тяжелое наследие» Microsoft определяет предвзятое отношение к языку C# у разработчиков свободного ПО.

Если же обратиться к компилируемым языкам, то в мире Linux безраздельно царят C и C++. Именно на них реализованы все популярные GUI-инструментарии, несмотря на то, что эти языки плохо приспособлены для программирования графических приложений. В результате разработчики библиотек вынуждены тем или иным способом обходить ограничения этих языков.

Создатели Qt пошли по пути модификации языка C++, дополнив его системой сигналов и слотов и метаобъектной информацией. Строго говоря, программы, использующие Qt, написаны не на C++, а на некоем особом языке, который сначала транслируется в C++ с помощью метаобъектного компилятора (MOC).

Библиотека GTK+ написана на стандартном C, который еще менее удобен для GUI-приложений, чем C++. Чтобы внести в GTK+ объектно-ориентированные абстракции и информацию о типах объектов, была создана библиотека GObject (<http://library.gnome.org/devel/gobject/stable/>). Использовать GObject в C напрямую сложно и неудобно, однако эта библиотека позволяет легко создавать стандартизированные объектно-ориентированные привязки [bindings] для других языков программирования. Желание использовать готовую объектную систему GObject и удобный высокоуровневый синтаксис привели к появлению сразу двух новых языков — Vala и Genie, о которых и пойдет речь в этой статье.

Идеология Vala и Genie

Создатели Vala и Genie довели идею, заложенную в Qt, до логического завершения. Раз уж нужно дополнять базовый язык новыми конструкциями, то лучше просто создать новый язык более высокого уровня с максимально удобным синтаксисом. Программы, написанные на Vala и Genie, сначала транслируются в стандартный C, а затем компилируются. Объектная модель языков базируется на GObject, скрытой в недрах компилятора.

Vala и Genie часто позиционируются как языки, облегчающие написание программ для Gnome. Может создаться впечатление, что они — сугубо специализированные. Это совсем не так! Vala и Genie могут использоваться в любой системе, куда портирована библиотека Glib (т.е. практически везде), а графический интерфейс на них можно создавать везде, где есть библиотеки GTK+. Компилятор Vala прекрасно работает в Linux, Mac OS X и Windows, что делает этот язык удобным средством для разработки кроссплатформенных GUI-приложений с помощью GTK+. У компилятора имеется режим POSIX-совместимости, в котором устраняется даже зависимость от Glib, но ценой потери некоторых возможностей языка.

Функционально оба языка практически идентичны и отличаются только синтаксисом. Vala очень похож на C#, тогда как Genie является своеобразной амальгамой Python, D и Delphi. Компилятор, утилиты и библиотеки являются общими для обоих языков. Компилятор различает тип синтаксиса по расширению файлов — **.vala** и **.gs**, соответственно.

Программы на Vala превращаются в код на C и поэтому непосредственно компонируются с внешними библиотеками без дополнительных «оберткок». Для использования любой библиотеки в Vala достаточно создать интерфейсный файл ***.vapi** с инструкциями для компилятора. Библиотеки, написанные на Vala и Genie, имеют автоматически сгенерированные заголовочные файлы и также могут сразу же использоваться в C/C++.

В отличие от C# и Java, Vala — настоящий компилируемый язык (хотя и с предварительной стадией трансляции кода). Производительность и расходы памяти у программ на Vala мало отличаются от изначально написанных на C.

Vala предоставляет все возможности современных объектно-ориентированных языков и многие удобные конструкции: классы и интерфейсы, свойства, сигналы, цикл foreach, лямбда-функции, неявную типизацию [type inference], обобщенные классы [generics], автоматическое или «ручное» управление памятью, исключения, контрактное программирование и регулярные выражения как часть языка.

Библиотеки, IDE, проекты

Язык Vala очень молод, но несмотря на это, он уже «оброс» впечатляющим инструментарием и даже обзавелся собственной IDE. В стандартной инсталляции Vala имеются привязки к десяткам библиотек (<http://live.gnome.org/Vala/BindingsStatus>). Среди них — все библиотеки GTK+, основные библиотеки платфор-

➤ Официальная страница языка Vala (<http://live.gnome.org/Vala>) подчеркивает его ориентацию на Gnome, хотя на самом деле он вполне универсален.



мы Gnome, а также *alsa*, *gsl*, *webkit*, *zlib*, *sqlite*, *OpenGL* и многие другие. В Vala имеется собственная библиотека обобщенных контейнеров *Gee* (<http://live.gnome.org/Libgee>), напоминающая STL в C++ (подробнее о ней ниже).

Поддержка Vala имеется в нескольких популярных IDE (MonoDevelop, Anjuta, Eclipse), но самой интересной является IDE Val(a)ide, написанный целиком на Vala.

Val(a)ide очень проста, но содержит практически все необходимое для комфортной работы с Vala. Имеется поддержка проектов, подсветка синтаксиса, встроенная система сборки наподобие Make. Очень удобно реализовано подключение к проекту нужных библиотек. Единственным существенным недостатком Val(a)ide является отсутствие интегрированного отладчика, хотя компилятор Vala поддерживает отладку через стандартный Gdb. Поскольку Val(a)ide не достигла даже версии 1.0, отладчик в ней должен еще появиться.

Несмотря на молодость языка, на Vala написаны десятки программ (http://live.gnome.org/Vala/#Projects_Developed_in_Vala). Vala проник и в область «дистрибутивостроения» – на нем базируется инсталлятор Paldo Linux, а на Genie – многие утилиты для Puppy Linux (см. **LXF132**).

В Ubuntu Vala устанавливается из ppa-репозитория. Достаточно ввести в консоли следующие команды:

```
sudo add-apt-repository ppa:vala-team
sudo apt-get update
sudo apt-get install valide-dev libgee-dev
```

Это установит библиотеку контейнеров *Gee*, IDE Val(a)ide, а также сам компилятор и все нужные файлы в качестве зависимостей.

В других системах Vala устанавливается из репозитория или исходных кодов, как описано на официальном сайте (<http://live.gnome.org/Vala/#Download>). Процесс сборки компилятора прост, так как единственной зависимостью является GLib версии 2.14 или выше.

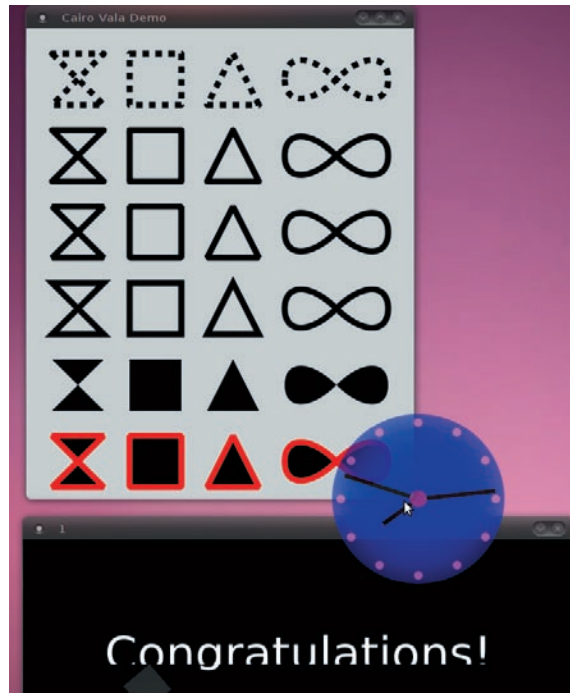
Первая программа

Традиционно мы напишем простейшую программу «Hello world!», но не в совсем скучном консольном варианте, а с использованием GUI на GTK+ и в двух вариантах – для Vala и Genie. Официальная документация по Genie очень скудна, поэтому дополнительно использовались сторонние ресурсы (<http://bkhome.org/genie/index.html>). В Genie для структурирования кода используется отступы (как в Python). Они создаются табуляциями, или пробелами, если в начале файла имеется директива [*indent=число пробелов*]. Начнем с варианта на Vala:

```
hello.vala:
using Gtk;
int main (string[] args) {
    Gtk.init (ref args);
    var window = new Window (WindowType.TOPLEVEL);
    window.title = "First GTK+ Program";
    window.set_default_size (300, 50);
    window.position = WindowPosition.CENTER;
    window.destroy.connect (Gtk.main_quit);
    var button = new Button.with_label ("Click me!");
    button.clicked.connect ((source) => {
        source.label = "Thank you";
    });
    window.add (button);
    window.show_all ();
    Gtk.main ();
    return 0;
}
```

Скомпилировать программу можно командой

```
valac --pkg gtk+-2.0 hello.vala
```



► Примеры программ на Vala (http://live.gnome.org/Vala#Sample_Code), использующие привязки к библиотекам Cairo и Clutter (**LXF132**).

Выполнение любой программы в Vala начинается с функции **main()**, принимающей массив строк, содержащий параметры командной строки. Сначала, с помощью **Gtk.init(ref args)**, инициализируется GTK+. Директива **ref** указывает, что параметр передается по ссылке, а не по значению. Далее создается объект Window с использованием стандартных констант GTK+:

```
var window = new Window (WindowType.TOPLEVEL);
```

Возникает впечатление, что переменная window создается без указания типа. На самом деле здесь срабатывает механизм неявной типизации для локальной переменной [type inference] по правой части оператора присваивания. Можно было бы объявить тип переменной явно:

```
// Явная типизация в Vala
```

```
Window window = new Window (WindowType.TOPLEVEL);
```

но для сложных типов, особенно обобщенных, неявная типизация становится просто спасительной. Сравните, например, эти два объявления в Vala:

```
MyFoo<string, MyBar<string, int>> foo = new MyFoo<string, MyBar<string, int>>();
```

```
var foo = new MyFoo<string, MyBar<string, int>>();
```

Первое читается с трудом, тогда как второе (с неявной типизацией) гораздо понятнее.

Далее задаются свойства окна и назначается функция-обработчик для сигнала окна destroy – в данном случае это **Gtk.main_quit**:

```
window.destroy.connect (Gtk.main_quit);
```

Эта лаконичная запись намного проще и понятнее жутковатого кода на чистом C в GTK+ и даже макроса **connect()** в Qt.

Далее создается кнопка и назначается обработчик ее сигнала clicked. Вместо применения отдельной функции-обработчика здесь «на лету» создается безымянная лямбда-функция. Конструкция

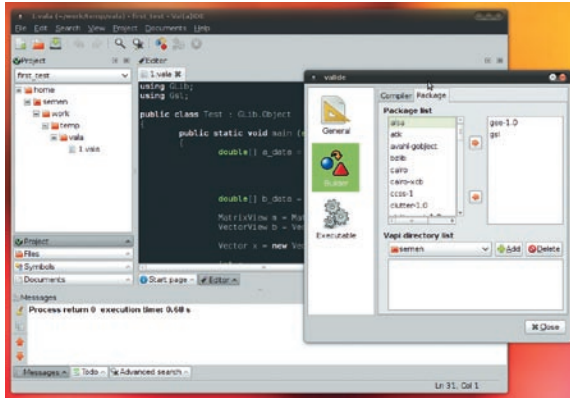
```
(source) => { source.label = "Thank you"; }
```

возвращает объект, являющийся функцией одной переменной (source), а именно такую сигнатуру ожидает сигнал clicked. Тип переменной указывать не нужно: он определяется компилятором автоматически и является ссылкой на объект, вызвавший сигнал, то есть нашей кнопкой.

Наконец, мы добавляем кнопку в окно, показываем его содержимое функцией **show_all()** и входим в цикл обработки событий

»

› Интегрированная среда разработки для Vala и Genie Val(a)ide сама полностью написана на Vala.



GTK+ с помощью `Gtk.main()`. Выход из этого цикла осуществляется только после закрытия окна программы.

Та же программа на языке Genie выглядит еще более лаконично:

```
hello.gs:
uses
    Gtk
init
    Gtk.init (ref args)
    var test = new TestWindow ()
    test.show_all ()
    Gtk.main ();
class TestWindow : Window
    init
        title = "Test Window"
        set_default_size (300, 50)
        window_position = WindowPosition.CENTER
        destroy += Gtk.main_quit
        var button = new Button.with_label ("Click Me")
        button.clicked += def (btn)
            btn.label = "Thank you!"
        add (button)
Компилируется она аналогичным образом
valac --pkg gtk+-2.0 hello.gs
```

Выполнение программы в Genie начинается с блока `init`, находящегося вне определений классов и функций. В отличие от предыдущего примера, используется немного другой подход – мы создаем класс `TestWindow`, унаследованный от стандартного `Window`. В конструкторе класса (блок `init`) выполняются те же действия, что и в предыдущем примере. Синтаксис присоединения функций-обработчиков к событиям в Genie еще более прост, чем в Vala, и использует оператор `+=`:

```
destroy += Gtk.main_quit
Объявление лямбда-функции также на вид отличается, но по сути – все в точности как в Vala:
button.clicked += def (btn)
    btn.label = "Thank you!"
```

Явное объявление переменных в Genie выглядит по-своему, так как используется синтаксис Pascal/Delphi с указанием типа после переменной:

```
int_val: int = 10
double_val: double = 12.5
button: Button = new Button.with_label ("Click me!")
```

В основном блоке `init` мы создаем экземпляр нового класса, показываем элементы окна функцией `show_all()` и входим в цикл обработки событий.

Еще один нетривиальный момент в обоих примерах – это отсутствие операторов `delete` после динамического создания объектов оператором `new`. Vala и Genie – языки с автоматическим управлением памятью, однако, в отличие от Java, предусмотрено «ручное» управление с помощью указателей, если необходимо.

Особенности языков

Полностью описать синтаксис двух языков в короткой статье невозможно, но к счастью, это и не нужно. Все конструкции языка Vala будут понятны любому программисту, знакомому с C++, C# или Java, а синтаксис Genie отличается от стандартного Python только в деталях. В то же время некоторые особенности этих языков заслуживают упоминания.

› Строки

Все строки в Vala и Genie имеют формат UTF-8. Тип `string` задает строку произвольной длины, которая является неизменяемым [immutable] объектом. Строку можно объявить как стандартным образом, так и в стиле «строки в тройных кавычках», знакомом по языку Python:

```
string text = "Обычная строка";
string str = """Строка в тройных кавычках ("verbatim string")".
В таких строках не обрабатываются специальные символы \n, \t, \ и т.д. Они могут содержать кавычки и занимать несколько строк.""";
```

Строки, предваряемые символом `@`, являются «строковыми шаблонами». В них можно вставлять переменные и выражения с помощью префикса `$(`, как это делается во многих скриптовых языках. Значения встроенных в строку переменных и выражений автоматически преобразуются к строковому типу. Удобство такого способа форматирования строк трудно переоценить:

```
int a = 10, b = 20;
string str = @"$a * $b = $(a * b)"; // Результирующая строка:
"10*20 = 200"
```

Получить подстроку или отдельный символ можно с помощью «срезов». Отрицательные значения индексов отсчитываются от конца строки:

```
string greeting = "hello, world";
string s1 = greeting[7:12]; // => "world"
string s2 = greeting[-4:-2]; // => "or"
unichar c = greeting[7]; // => 'w'
```

Поддерживаются также многие стандартные строковые функции библиотеки `GLib`.

› Массивы

Массивы в Vala могут быть динамическими (созданными в куче) и статическими (созданными в стеке).

```
int[] a = new int[10]; // Динамический массив
int f[10]; // Статический массив
```

Для динамических (но не для статических) массивов поддерживается добавление элементов «на лету»:

```
int[] e = {}
e += 12;
e += 5;
e += 37;
```

Нужно иметь в виду, что такая возможность рассматривается исключительно как еще один способ инициализации – стандартные массивы не предназначены для использования в качестве контейнеров с переменным количеством элементов. Для этих целей лучше использовать классы-коллекции из библиотеки `Gen` (массивы, списки, хэши и т.п.).

```
Поддерживаются также многомерные массивы
int[,] c = new int[3,4];
int[,] d = {{2, 4, 6, 8},
            {3, 5, 7, 9},
            {1, 3, 5, 7}};
d[2,3] = 42;
```

› Управляющие конструкции

Синтаксис управляющих конструкций `if`, `for`, `while`, `switch` в Vala полностью аналогичен синтаксису C/C++, а в Genie очень похож

на синтаксис Python. Отдельного внимания заслуживает только конструкция `foreach`, позволяющая легко перебирать элементы массивов и коллекций:

```
foreach (int a in int_array) { stdout.printf("%d\n", a); }
```

В Genie в качестве `foreach` выступает разновидность цикла `for`:

```
for s in args do print s
```

Тип переменной `s` компилятор определяет автоматически.

» Делегаты

В Vala и Genie нет указателей на функции. Если нужно передать одну функцию в виде параметра другой, используются делегаты:

```
delegate void DelegateType(int a);
void f1(int a) {
    stdout.printf("%d\n", a);
}
void f2(DelegateType d, int a) {
    d(a); // Вызов делегата с переданным параметром
}
void main() {
    f2(f1, 5); // Функция f1 передается в параметре-делегате
}
```

Код с делегатами читается куда лучше, чем с указателями в стиле C/C++.

» Обобщенные классы

Vala и Genie поддерживают обобщенные классы, то есть классы, способные работать с разными типами данных. Конкретный тип указывается при создании экземпляра класса. Обобщенные классы [Generics] не являются полным аналогом шаблонов [Templates] в C++; единственное их назначение – избежать написания одного и того же кода для нескольких типов данных, которые обрабатываются одинаковым образом. Обобщенные классы не предназначены для шаблонного метапрограммирования, как в C++. Синтаксис обобщенных классов в Vala похож на шаблоны в C++, но более прост. Имя обобщенного типа задается в угловых скобках:

```
public class Adder<G> : GLib.Object {
    private G data;
    public void do(G a, G b) {
        this.data = a+b;
    }
    public G get_sum() {
        return this.data;
    }
}
var v = new Adder<int>();
```

В Genie имя обобщенного типа задается конструкцией «`имя_типа`»:

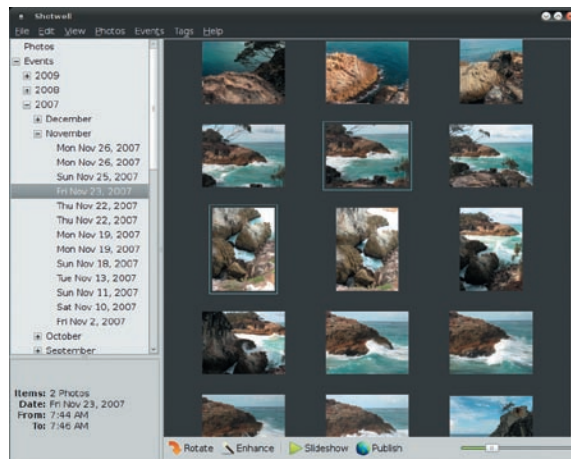
```
class Adder of G : Object
    _data : G
    def do(a: G, b: G)
        _data = a+b;
    def get_sum(): G
        return _data
var v = new Adder of int
```

Переменные, начинающиеся со знака подчеркивания, автоматически получают атрибут `private`.

» Контрактное программирование

Vala поддерживает базовые конструкции контрактного программирования `requires` и `ensures`:

```
double method_name(int x, double d)
    requires (x > 0 && x < 10)
    requires (d >= 0.0 && d <= 1.0)
    ensures (result >= 0.0 && result <= 10.0)
{
```



» Менеджер фотографий Shotwell для среды Gnome (<http://yorba.org/shotwell/>) написан на Vala и является достойной альтернативой тяжеловесному F-Spot (стр. 33).

```
return d * x;
```

Специальная «магическая» переменная `result` имеет тип, возвращаемый функцией.

» Библиотека обобщенных контейнеров Gee

Библиотека *Gee* в Vala выполняет ту же функцию, что и *STL* в C++. В *Gee* имеются структуры данных и итераторы, но, в отличие от *STL*, нет алгоритмов. Синтаксис шаблонных классов *Gee* очень похож на синтаксис C++, так что программисты, привыкшие к шаблонным контейнерам в C++, будут чувствовать себя как дома. *Gee* содержит несколько классов-коллекций, самыми часто используемыми из которых являются вектор `ArrayList<G>`, список `LinkedList<G>` и ассоциативный контейнер `HashMap<K,V>`. Контейнеры *Gee* намного сильнее интегрированы в базовый язык, чем контейнеры *STL*. Например, любой контейнер можно использовать в циклах `foreach`:

```
var list = new ArrayList<int> ();
list.add (1); // Добавление элементов
list.add (2);
list.add (4);
list.insert (2, 3); // Вставка элемента
list.remove_at (3); // Удаление элемента
foreach (int i in list) { // Интеграция в цикл foreach
    stdout.printf ("%d\n", i);
}
list[2] = 10; // Доступ по индексу
```

Проверить наличие элемента в любом контейнере можно оператором `in`:

```
if (2 in list) { // Аналогично list.contains(2)
    stdout.printf ("Двойка в списке есть!\n");
}
```

В итоге

Появление любого нового языка программирования обычно вызывает опасения, что он окажется очередным нежизнеспособным курьезом. Vala и Genie благополучно избежали этой участи. Эти языки базируются на стабильной и очень популярной библиотеке *GLib*, но позволяют программисту работать с современным компилируемым языком высокого уровня, при полной совместимости с C. Vala и Genie отлично приспособлены для написания GUI-приложений с использованием *GTK+* и кардинально облегчают работу с этим графическим инструментарием. Можно сказать, что Vala – это «C# без виртуальной машины», а Genie – «компилируемый Python». Производительность этих языков близка к чистому C, а удобство синтаксиса и богатство конструкций – к скриптовому языку. Vala и Genie будут достойным дополнением арсенала любого программиста, особенно работающего с *GTK+*. **LXF**

Что за штука... Qt Quick?

Бросок *Clutter* к власти был резок, но выбить Nokia с рынка интерфейсов для мобильных устройств не так-то просто, считает Пол Хадсон.

» Стоп-стоп... я сроду не слышал ни о каком *Clutter*. К власти, говорите, бросался?

Clutter такая штука, что о нем не обязательно слышать – он работает на заднем плане; но работает грандиозно!

» Что же это такое?

Это пакет инструментов для работы с графикой: с его помощью можно рисовать на экране фигуры и заставлять их двигаться.

» Не вижу ничего особенного.

А надо бы! *Clutter* работает через OpenGL, который выдаёт удивительно плавную графику даже на маломощном оборудовании. Заставить объекты двигаться не так-то просто, а *Clutter* запросто справляется с анимацией: масштабирует, вращает и двигает фигуры на экране.

» И рвался к власти?

О да! В Ubuntu Netbook Remix *Clutter* используется для поддержки чудных интерфейсов

» А на что его заменили?

Предсказуемо – на *Qt*, собственную технологию Nokia. Правда, есть некоторая загвоздка: *Qt* прекрасно работает на Mac, Windows, Linux и даже мелких устройствах Symbian, но остаётся традиционным набором интерфейса пользователя – с кнопками, меню, ползунками и флажками. В *Clutter* всё другое: просто грузится картинка, грузится текст, и всё это движется. Поэтому интерфейсы на основе *Clutter* выглядят словно созданные художниками: сплошные картинки и мультики.

» Да чем плох стандартный набор? Мне нравится единообразие интерфейсов в приложениях.

Здесь есть свои «за» и «против». Конечно, со стандартным инструментарием все приложения будут похожими – но почему, скажем, *F-Spot* должен быть похож на *OpenOffice.org*? Работа в *Clutter* всегда начинается с чистого листа: вы добавляете свои объекты и заставляете их делать что вам надо. С *Qt* вы просто пользуетесь – ну, или пользовались – стандартными виджетами *Qt*. Даже самые распрекрасные темы всё равно остаются стандартными виджетами *Qt*.

» Здесь-то и появляется *Qt Quick*?

Да! *Qt Quick* наследует классные идеи *Clutter* и внедряет их в жизнь. Поэтому в качестве основы программист получит виджеты – прямоугольники, изображения и текст – а ещё таблицы, видео, жесты и даже встроенный web-браузер на базе *WebKit*.

» Нужна ли нам ещё одна библиотека пользовательского интерфейса? Меня давно уж бесит, что *GTK*-приложения плохо вписываются в мой рабочий стол *Qt*.

Тут-то и начинается интересное: *Qt* присутствует в названии потому, что *Qt Quick* на *Qt* и по-

строен – ему не надо загружать особые библиотеки, и на фоне *Qt*-приложений он выглядит вполне уместно. По сути, *Quick* – отделочный слой, который использует начинку *Qt* и придаёт ему новый шарм.

» Так это всего лишь *Qt*? А шума-то!

Вам всё станет ясно, когда вы разберётесь, из чего состоит *Qt Quick*. Как я и сказал, *Qt* составляет его основу: он кроссплатформенный, он обладает всей мощью рендеринга *Qt*, и при желании вы можете воспользоваться всем функциональным богатством *Qt*. Вторая важная составляющая – *Qt Creator*, конструктор интерфейсов для *Qt*, с которым интерфейс можно создавать, просто перетаскивая элементы. *Quick* добавил лоска, но основная идея не меняется: вы накидываете в интерфейс изображения и прочие штуки, а затем наделаете их функциональностью.

» Вроде последней версией *Qt* была 4.6.

» Что нового? Где там вставлен *Quick*?

Третья опора треножника *Quick* называется *QML*, что означает *Qt Meta-Object Library* и звучит страшнее, чем является по сути. А если попросту, *QML* позволяет создавать пользовательские интерфейсы с помощью простых текстовых описаний объектов вашего окна. Почти как при создании CSS: чтобы определить конфигурацию, размер, цвет и прочие атрибуты объекта, достаточно описать его «правильными» словами. Естественно, в окне конструктора *Qt Creator* всё это генерируется за вас. Остаётся добавить JavaScript-подобный код, чтобы получить необходимую функциональность.

» Ну и что здесь революционного? В *GTK* тоже создают интерфейсы с помощью XML. По мне, текст везде текст.

Верно, но как насчет возможности подцепить функциональность прямо к коду *Quick*?

«Qt Quick наследует классные идеи Clutter и внедряет их в жизнь.»

сов на дешевых чипсетах Intel, применяемых в нетбуках. На системах Moblin *Clutter* применяется по тем же причинам: весь функционал OpenGL можно реализовать на оборудовании с низкой производительностью.

» Минуточку... не в этом ли журнале я читал, что *Clutter* убрали из последней версии Moblin в этом... как его?

MeeGo. Вы правы, Intel и Nokia решили объединить проекты Moblin и Маемо, и *Clutter* остался не у дел. Честно говоря, жаль: *Clutter* успел собрать довольно мощную «группу поддержки».

» Жуть кошмарная! Пару лет назад я пробовал Qt, но C++ меня доконал.

Согласен, Qt порой бывает жутковат. Точнее, частенько бывает, ха-ха. Но, как я уже говорил, Quick – дело иное. Здесь используется инструмент QtDeclarative (ещё одно не в меру устрашающее название). Он действует так: вы не описываете действия объектов, а «говорите» объектам, чего хотите от них, а они уж сами «думают», как это сделать.

» Я уже сто лет так делаю! Говорю: «Хватит жрать ОЗУ, Firefox!» – а он не слушает...

Увы, заклинания против ПО бессильны. А вот QtDeclarative работает: вы говорите, куда вставить объекты, и там они и размещаются. Если надо, чтобы объект переворачивался при наведении курсора, достаточно указать исходное (лицевая сторона) и итоговое состояние (изнанка) и связать это преобразование с событием – наведением курсора. Анимацию QtDeclarative берёт на себя. Объекты

можно двигать, вращать, масштабировать, перекрашивать – можно даже задать время анимации или по ходу менять её скорость.

» И вправду просто. Причём никакого C++ – это вообще отлично!

C++ никуда не делся: ведь Quick базируется на Qt, а тот, в свою очередь, на C++; но C++ упрятан вглубь. И это неплохо: если вам мало функций, предлагаемых Quick, можно вернуться к связке C++ и Qt. Qt готов к использованию: при желании можно конвертировать всё приложение в стандартный Qt – а можно добавлять его кусочками там и тут, по вкусу. В Qt есть класс C++ под названием QtDeclarativeView, и с его помощью можно читать QML и произвольно манипулировать им.

» О нет, C++ лучше не трогать, упрятали – и ладно.

Главное, что у вас есть выбор. Хотите пользоваться QtDeclarative – пожалуйста. Хотите спуститься

в C++ – к вашим услугам. А можно их сочетать – тоже работает; решать вам.

» Так, руки уже чешутся попробовать. Что надо запустить?

Большой недостаток Quick – он будет доступен только в Qt 4.7, который сейчас находится в стадии разработки у Nokia. Есть основания надеяться, что выпуск состоится в ближайшие несколько недель, тогда и попробуйте. Qt 4.7 включает Qt Creator 2.1 с поддержкой QML – и вы сможете узнать, как стряпаются интерфейсы перетаскиванием.

» А можно почитать об этом, чтобы слегка подготовиться?

Конечно! Лучше всего начать с <http://doc.qt.nokia.com/4.7-snapshot/declarativeui.html> – там описаны различные функции, собрано несколько руководств и даже приведено несколько примеров сопряжения QML и C++. **LXF**

Разработчики и дизайнеры! Вот новый способ создания интерфейсов и приложений!

Qt Quick

Не надо знать C++!

Кодовая база сохраняется!

Это просто!
Это быстро!

Вы ограничены только своей фантазией!

Qt

QML



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.

Откроем «открытое ПО»

На прошлой неделе мне удалось на несколько дней попасть в Брюссель на Саммит разработчиков Ubuntu (Ubuntu Developer Summit). Он проводился в бельгийском лесу. Ну, то есть, в довольно неплохой гостинице в бельгийском лесу. На пеньках нам сидеть не пришлось. Разработчикам Ubuntu – во многих отношениях, ядру сообщества Ubuntu – которые в основном переписывались через Интернет, предоставилась бесценная возможность пообщаться вживую. Здесь также начали вырисовываться контуры следующего релиза Ubuntu, **Maverick Meerkat**.

Настоящие члены этого сообщества говорят на своем особом языке. Вот его реальные примеры, которые я там услышал: «Я работаю в пространстве ARM».

«Нам нужно синхронизироваться вверх по течению» (Будет ли это контрпродуктивным, если я кину им спасательный жилет?)

Наверное, если не понимаешь этого, то и нечего там делать.

Процесс вовлечения

Как преподавателю, писателю и радетелю за чистоту английского языка мне было далеко до этого сообщества, и я играл обескураживающе пассивную роль в происходящем. Но даже если кое-что и ставило меня в тупик, я был поражен тем, насколько открытым и простым было мероприятие. Принять в нем участие мог любой желающий, а для тех, кто не смог добраться, организовали общение с коллегами посредством IRC-чатов и аудиопотоков. Где еще можно было бы обсудить, например, поддержку multi-touch для рабочего стола или принять участие в дискуссии о заплатках ядра по безопасности, прямо с исходным кодом? (Если вам в самом деле хочется углубиться в подобные вещи, зайдите на <https://wiki.ubuntu.com/SecurityTeam/Roadmap/KernelHardening>).

Уверен, что в Редмонде такие дискуссии проходят только за закрытыми дверями.

Добротное администрирование систем из причудливых заворотов кишок серверной.



Вскрытие файловой системы

Sleuthkit

 Долой лупу и порошок для определения отпечатков пальцев – переходим на утилиты.

Smart (Self-Monitoring Analysis and Reporting Technology) – технология самоконтроля, анализа и оповещения, разработанная ведущими производителями жестких дисков для предоставления подробной диагностической информации о состоянии накопителя. Идея тут в том, чтобы повысить надежность хранения, определяя приближение сбоя до того, как он произойдет и вы пострадаете от простоя диска или потери данных. Эта идея не нова. В «Космической одиссее: 2001», как вы помните, HAL 9000 предсказал сбой в работе параболической антенны корабля АЕ35.

Мой друг ездит по клиентам, собирая компьютеры для судебного анализа. Хорошую компанию ему мог бы составить *Sleuthkit* (www.sleuthkit.org) – набор утилит для детальной проверки файловой системы на компьютерах, которые были атакованы или могут содержать «неподходящую» информацию. Утилиты работают с файловыми системами NTFS, FAT, HFS+, Ext2, Ext3, UFS1 и UFS2. Пакет *Sleuthkit* содержит библиотеку на C и около 25 утилит командной строки и map-страниц. *Autopsy* – графическая оболочка для этого инструментария.



В клиенте *Autopsy* можно просмотреть удаленные файлы на диске или его образе.

Проверьте себя

Если вы хотите исследовать какую-нибудь файловую систему, но недавно атакованных под рукой нет, несколько подходящих образов вместе с задачами на исследование можно найти по адресу <http://old.honeynet.org/scans>. Это поможет вам скоротать время дождливыми летними днями.

Sleuthkit выполняет «мертвый анализ», когда система анализа используется для проверки диска или его образа из сомнительной системы. Альтернатива – «живой анализ», при котором сомнительная система проверяется во время своей работы. При этом *Autopsy* и *Sleuthkit* запускаются с компакт-диска в недоверенной среде. Это часто применяется во время «ответа на инцидент», когда инцидент подтверждается. После этого можно получить физический доступ к системе и выполнить «мертвый анализ». Используемые *Autopsy* термины отражают серьезную природу ее работы. В ней можно «открыть дело», назначить «следователей» и поместить результаты в «сейф с уликами».

Утилита выполняет текстовый поиск в образе файловой системы, даже во фрагментах удаленных файлов. В ней можно построить временную шкалу активности файловых систем, в том числе на нескольких компьютерах. Утилита проверяет контрольные суммы MD5 файлов, сравнивает их с контрольными суммами «доверенных файлов» в базе данных и ищет «плохие» файлы. *Sleuthkit* не автоматизирует процесс расследования, но предоставляет средства выполнения этой работы.

Контейнерные технологии

Широкий спектр В Linux таких множество – разберемся с их типами.

При каждом запуске программы в Linux она разделяет некоторые ресурсы с другими приложениями и виртуализирует (или изолирует) остальные. Какие ресурсы разделяются, а какие изолируются, зависит от используемой технологии. Вот небольшой обзор.

» **Процессы** Процессы – низший уровень контейнеризации, и вы даже, наверное, ворчите на меня за то, что я отнес их к этой категории. Но процесс – и в самом деле контейнер, в котором выполняется программа. Среди прочего он изолирует пространство памяти программы (у каждого процесса есть иллюзия, что все адресное пространство принадлежит ему) и наборы файловых дескрипторов программы (например, каждый процесс знает, куда подключен его стандартный поток вывода). Эта технология контейнеризации фундаментальна для Linux и Unix и существует с незапамятных времен.

» **Chroot** Идея здесь в том, чтобы ограничить программу определенной частью файлового дерева («тюрьмой» или «песочницей»). Механизм основан на системном вызове *chroot* (изменить корень), который изменяет представление процесса о том, где находится корневой каталог (*/*). Например, если процесс делает *chroot* в каталог */var/run/foobar* и затем обращается к файлу */etc/fstab*, на самом деле он будет обращаться к файлу */var/run/foobar/etc/fstab*.

Для правильно написанной программы это мера безопасности – программа не сможет получить доступ ни к каким файлам вне «тюрьмы». Часто с применением этой технологии работают демоны – на ум приходит DNS-сервер *Bind*. Но данный механизм нельзя назвать сильным, да он никогда и не претендовал на это. Если демон выполняется под пользователем *root*, и у него есть уязвимость (к примеру, ошибка при переполнении буфера), существует несколько хорошо документированных способов вырваться из тюрьмы. Механизм *chroot* на самом деле предназначался для того, чтобы предоставить программе собственное видение файловой системы, а разработчикам – «песочницу» для тестирования новых программ, а не для того, чтобы сдерживать программы с уязвимостями.

При создании «тюрьмы» для программы нужен определенный подход. Нельзя просто бросить узника в камеру без еды и воды и ждать, что он выживет. В «тюрьме» должны быть файлы, необходимые для работы программы, файлы данных, файлы настройки; наверное, копии некоторых файлов из */etc* и */dev*, и т. д.

Resource→	Memory	Open file descriptors	Process Table	Network stack (e.g. ports)	File system	Kernel and CPU
Container Technology ↓						
Process	Separate	Separate	Shared	Shared	Shared	Shared
Chroot jail	Separate	Separate	Shared	Shared	Separate	Shared
Linux container	Separate	Separate	Separate	Separate	Separate	Shared
Kvm, xen, etc	Separate	Separate	Separate	Separate	Separate	Separate

» **Контейнеры** Контейнеры в Linux – самая новая, самая гибкая и (на данный момент) самая страшная технология контейнеризации. Контейнеры очень гибки: они позволяют управлять тем, какие ресурсы нужно изолировать, а какие – разделить. Контейнеры виртуализируют (изолируют) как минимум идентификаторы процессов, механизмы IPC «System V» и точки монтирования (которые являются представлениями контейнера о файловой системе).

Кроме этого, задав соответствующую конфигурацию контейнера, можно изолировать другие ресурсы, такие как имя хоста (контейнера может иметь другое имя хоста по сравнению с основной системой) и сетевые

интерфейсы (например, у контейнера могут быть свои виртуальные сетевые интерфейсы, каждый со своим MAC-адресом). Контейнеры могут разделять с основной системой некоторые части файловой системы и иметь собственное представление других.

Можно создать контейнер для запуска отдельной программы или системы целиком. Как ни странно, второе проще – меньше работы по выбору и настройке разделяемых ресурсов – системы.

Технология контейнеров опирается на возможности, появившиеся только в самых последних ядрах – для полной функциональности нужна версия 2.6.29 или больше. И технология в целом на данный момент выглядит сыровато. Есть несколько утилит для управления контейнерами, но ничего (насколько я смог найти) – для автоматизации создания и настройки контейнеров. Например, в *Virt-manager* нет мастера создания новых виртуальных машин. Думаю, это вопрос времени. Если вы не можете ждать и готовы принять на себя боль настройки контейнера вручную, взгляните на руководство Бодхи Зазена [Bodhi Zazen] по адресу <http://blog.bodhizazen.net/linux/lxc-configure-ubuntu-lucid-containers>.

» **Виртуальные машины** Виртуальные машины – предельная технология контейнеризации, конечно, не считая физических компьютеров, на которых можно запускать нужные программы. Внутри виртуальной машины у вас появляется иллюзия, что у вас есть целый компьютер. У виртуальных машин есть собственное ядро ОС; на них можно (и так часто делают) запускать разные ОС.

Родные технологии Linux здесь – это *KVM* и *Xen*; доступны и некоторые проприетарные продукты, такие как *VirtualBox* от Sun и *VMware*. Тема виртуализации и виртуальных машин была довольно подробно раскрыта в последних номерах журнала включая *LXF132*, и здесь я не буду углубляться в нее.

» **Контейнеры характеризуются ресурсами, которые они виртуализируют и разделяют.**

«Существует несколько способов вырваться из тюрьмы.»

Вложенные контейнеры

Технологии контейнеризации могут вкладываться друг в друга, что позволяет создавать контейнеры внутри контейнеров. Все приложения выполняются внутри процессов. Процессы можно запускать в «тюрьмах» *chroot* или в контейнерах. Уж не знаю, зачем запускать «тюрьму» *chroot* внутри контейнера, потому что контейнеры сделают все то же самое, что и «тюрьма» *chroot*. И, конечно, контейнеры можно создавать на виртуальных машинах. Конечно, все эти контейнеры в конечном счете соперничают друг с другом за физические ресурсы компьютера – память, циклы процессора, ширину полосы пропускания сети и дисковое пространство.

Созерцающая Landscape

Landscape Управляйте несколькими компьютерами с Ubuntu через интуитивный web-интерфейс – если вас устраивает, что он не вполне свободен.

Администраторы больших сетей хорошо знают, что обновлять ПО на каждом компьютере по отдельности, подходя к нему и вставляя туда диск, невозможно – не хватит никакого времени. Они не могут позволить себе задач, усилия на решения которых линейно возрастали бы с ростом числа компьютеров. Необходимо какое-то централизованное масштабируемое решение. Если на компьютерах установлена Ubuntu, то таким решением может стать сервис *Landscape* от Canonical.

Для работы *Landscape* на каждый компьютер, которым вы хотите управлять, нужно установить небольшую агентскую программу (пакет *landscape-client*). Эти агенты периодически связываются с центральным сервером *Landscape* и сообщают ему состояние процессов, информацию о программах и аппаратной части, использовании ресурсов и т.д. и получают из очереди запросы на выполнение обновлений пакетов или на запуск скриптов. Сервер *Landscape* общается с каждым из управляемых узлов, администратор работает с сервером через web-интерфейс.

Изначально *Landscape* мог работать только на серверах Canonical; сейчас сервер можно взять свой. Canonical называет это *Dedicated Server Edition* (решение с выделенным сервером). Оно может представлять интерес для компаний, политика безопасности которых не разрешает передавать данные по публичным сетям, или для установок с ограниченным интернет-соединением. Но не пытайтесь найти пакет *landscape-server* в репозитории своего дистрибутива – его там нет. Исходный код сервера не является открытым (см. «Получение учетной записи» ниже).

С помощью *Landscape* можно получить список оборудования компьютеров, список процессов, запущенных на них (и завершить заблудшие процессы), построить графики средней загрузки, использования памяти, файла подкачки и дискового пространства. Эта информация пригодится при планировании характеристик оборудования. Можно даже написать собственные скрипты

оболочки для построения произвольных графиков (вероятно, по типу модулей расширения *Munin* от пользователей, для построения графиков использования ресурсов). Также можно создавать новые учетные записи пользователей. Но, пожалуй, самая важная функция *Landscape* с точки зрения сисадмина – возможность автоматизации установки, обновления и удаления пакетов на управляемых компьютерах.

Так как у сервера *Landscape* есть доступ к репозиториям Ubuntu и списки установленного на компьютерах ПО, он знает, когда нужно его обновить. Сначала нужно выбрать компьютеры, нуждающиеся в обновлении ПО, по отдельности или с помощью тэгов. Потом выберите пакеты, нажмите на кнопку «Apply Changes» [Применить изменения] – и готово! Кстати, пусть моя маленькая диаграмма архитектуры *Landscape* не сбивает вас с толку. Пакеты при установке или обновлении загружаются на управляемые компьютеры напрямую из репозитория, а не через сервер *Landscape*. Вы почти наверняка захотите установить кэш пакетов вроде *apt-proxy* или *Squid* или даже создать полное зеркало репозитория локальной сети с помощью *apt-mirror*, чтобы компьютеры не тратили трафик, загружая пакеты каждый раз.

Профили пакетов, новая возможность *Landscape*, появившаяся в версии 1.5, позволяют работать с целым набором пакетов как с одной

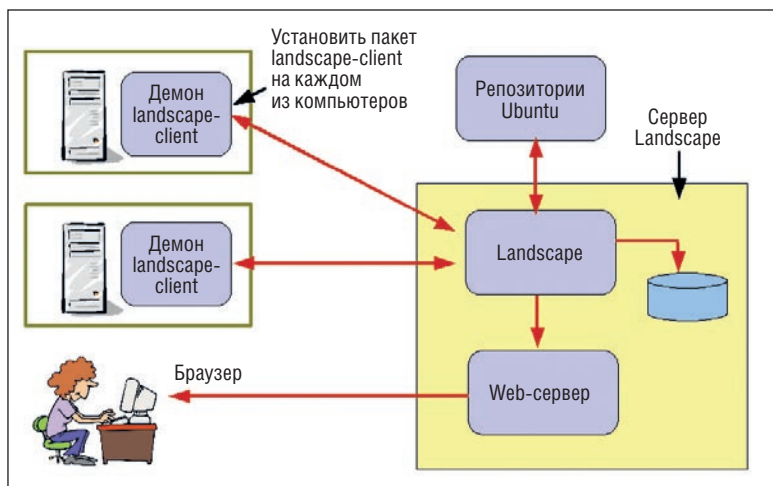
именованной сущностью. Скажем, нам нужно установить LAMP на новый сервер с Ubuntu. До появления профилей пришлось бы создавать отдельные наборы команд для установки *Apache*, *PHP* и *MySQL*. С появлением профилей нужно создать только профиль с одним именем, где были бы описаны эти зависимости. Потом профиль просто применяется ко всем компьютерам, у которых есть, например, тэг «lamp». Этот механизм обладает большой гибкостью и большими возможностями.

«Профиль применяется ко всем компьютерам с заданным тэгом.»

Предупреждения

Когда появится что-то, по мнению *Landscape*, требующее вашего внимания, он покажет «предупреждение». Это могут быть новые компьютеры, «ожидающие» приема в сеть, новые обновления пакетов или отсутствие ответа от того или иного компьютера в течение определенного времени. Предупреждения можно увидеть на домашней странице *Landscape* (Dashboard) после входа в систему. Также можно подписаться на предупреждения того или иного типа по электронной почте.

При выполнении установки или обновления пакетов или запуске скриптов на удаленных компьютерах нужно выбрать целе-



Управляемые компьютеры регулярно отчитываются перед сервером Landscape, где также есть web-интерфейс с панелью управления, за которой приглядывает администратор.

Получение учетной записи

Landscape – программа, ни бесплатная как пиво, ни свободная как слово. Это сервис Canonical, распространяющийся по подписке. Все контракты поддержки Canonical включают неограниченное использование сервиса *Landscape*. Также можно купить годовую подписку отдельно (www.canonical.com/contact/sales) или запросить 60-дневную бесплатную пробную версию (www.canonical.com/landscape/register).

вые компьютеры. Самый простой способ сделать это – выбрать их по одному с помощью галочек. Но тогда мы снова возвращаемся к схеме «усилия пропорциональны количеству компьютеров», которой тщились избежать. Поэтому каждому из компьютеров можно назначить текстовые метки (тэги) и производить действия с компьютерами, которые помечены определенным тэгом. Например, можно запустить обновление *Tomcat* на всех компьютерах с тэгом «webserver» или запустить скрипт обновления темы окошка входа в систему на всех компьютерах с тэгом «sales-desktp». Обычно тэги компьютера задаются при его начальной регистрации в *Landscape*. Тэг может описывать роль компьютера, его аппаратную часть или физическое расположение. Одному компьютеру можно назначить несколько тэгов: так, например, можно разом установить пакет на все четырехъядерные почтовые серверы в Бристоле.

Если вы хотите выполнять какие-то действия, которые выходят за рамки возможностей *Landscape*, можно написать собственные скрипты, сохранить их на сервере *Landscape* и применить их к заданному набору компьютеров. При создании скрипта вы задаете его имя, используемый интерпретатор, максимальное время выполнения и пользователя, под которым его нужно запустить.

Запуск скриптов

Конечно, потребуется ввести и сам код скрипта. Если ему нужны другие файлы, их можно добавить в виде вложений. Все это делается через web-интерфейс *Landscape*. Например, этот маленький скрипт установит в качестве фона рабочего стола изображение, переданное в качестве вложения:

```
cp $LANDSCAPE_ATTACHMENTS/moon.png $HOME/gconftool-2
--type string --set \
/desktop/gnome/background/picture_filename \
$HOME/moon.png
```

Вот более тонкий пример из одной строки, который выдает приглашение командной строки на управляемом компьютере:

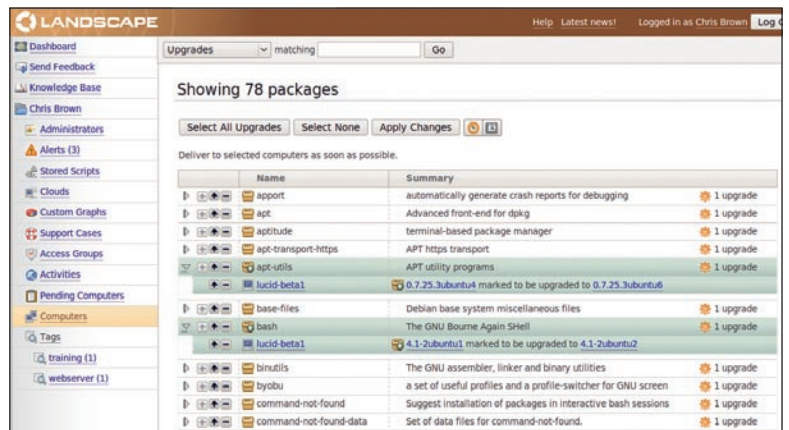
```
ssh -R 2048:localhost:22 foo.example.com
```

Эта команда создаст обратный туннель на **foo.example.com** на порте **2048**, который уйдет обратно на управляемый компьютер, и потом можно будет зайти с него по SSH:

```
ssh -p 2048 localhost
```

Для его работы между компьютерами должна быть настроена SSH-аутентификация.

Разрешение выполнения скриптов, несомненно, влияет на безопасность, и при регистрации компьютера в *Landscape* можно выбрать, разрешаете ли вы сервису это делать, и определить пользователей, запускающих скрипты. Предусмотрительно было бы ограничить скрипты пользователем nobody, но тогда скрипты будут ограничены правилом «смотри, но не трогай», потому что у пользователя nobody очень мало привилегий. Другой крайний случай – разрешить запуск скриптов от имени root. Однако (надеюсь, это не нужно прояснять, но я все равно проясню) в этом случае данные аутентификации *Landscape* становятся по сути отмычкой к домам на целой улице. И если вы разрешаете Canonical размещать сервер у себя, спросите себя, доверяете ли вы им безопасность их собственного сайта. Это просто напоминание о том, как обстоит дело в действительности, а не критика в адрес Canonical.



Вот как управляются компьютеры через *Landscape*:

- 1 Получить учетную запись на сервере *Landscape* (см. «Получение учетной записи» на предыдущей странице).
- 2 Установить клиента на каждый компьютер, которым вы хотите управлять. Для этого достаточно выполнить команду:


```
$ sudo apt-get install landscape-client
```
- 3 Зарегистрировать компьютер на сервере *Landscape* командой **landscape-config**.
- 4 Войти в свою учетную запись и одобрить ожидающий запрос на регистрацию.

Это проще, чем кажется. На самом деле, в установщике Ubuntu Server есть окно политики обновления (Upgrade Policy), в котором можно установить, настроить и зарегистрировать компьютер как клиент *Landscape* прямо во время установки. На третьем шаге обычная команда настройки *Landscape* может выглядеть так:

```
$ sudo landscape-config --computer-title=saturn \
--accout-name=fredflinstone \
--script-users=nobody,landscape,root \
--tags=webserver,training
```

Здесь **saturn** – имя компьютера, под которым он будет известен *Landscape*, параметр **--script-users** задает, от имени какого пользователя можно запускать скрипты на управляемом компьютере (через *Landscape*), а **--tags** содержит простые текстовые метки, используемые для идентификации группы компьютеров. **landscape-config** можно запустить и без параметров, тогда он сам запросит нужные.

После регистрации клиента зайдите на сервер *Landscape* (<http://landscape.canonical.com>), зайдите на вкладку Pending Computers [Ожидаящие компьютеры], выберите компьютер(ы) и нажмите Ассепт [Принять]. Пройдет некоторое время, прежде чем компьютер появится на экранах *Landscape* – помните, что сервер видит клиента только тогда, когда тот сообщает ему информацию.

Для более подробной информации зайдите на сайт www.youtube.com/user/CanonicalLandscape или на <http://help.landscape.canonical.com>. **LXF**

► Выбирайте обновления и обновляйте ПО на нескольких компьютерах за раз с помощью *Landscape*.

Landscape и облако

Облачные вычисления могут, как ни странно, повысить нагрузку на системного администратора. Физических компьютеров, с которых нужно вытирать пыль, может быть меньше, но «логических» компьютеров, на которых нужно устанавливать, настраивать и обновлять ПО, больше. *Landscape* может помочь управлять облаком Amazon EC2 и внутренними экземплярами *Eucalyptus*, за-

ставив их автоматически отображаться как зарегистрированные компьютеры.

Сначала зарегистрируйте облако в *Landscape* и предоставьте ему ключи доступа к web-сервисам Amazon или *Eucalyptus*. Существующие компьютеры в облаке должны появиться в *Landscape*. С помощью web-интерфейса *Landscape* можно также добавлять новые ком-

пьютеры, запускать и останавливать их. В AMI (Amazon Machine Image), который используется в облаке, должен быть установленный и настроенный **landscape-client**. Если его нет, управлять AMI через *Landscape* нельзя.

Я этого не пробовал, а документация создает впечатление о чем-то явно новом, но не ставшем обиходным. См. <https://help.landscape.canonical.com/ManagingEC2>.

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





Наши эксперты помогут вам с любым приложением Linux!



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

Как осчастливить человечество?

Что я могу сделать для людей? – вскричал Данко... а потом упал и умер.
Вольный пересказ пересказа Максима Горького сказки старухи Изергиль.

Н е секрет, что (надеюсь, только пока) образование по профилю «информатика» (она же – «компьютерные науки») в России фактически отсутствует. У нас есть пока ещё достаточно качественное фундаментальное образование по точным наукам, к которым элементы информатики как бы пристёгиваются. Также не секрет, что значительная часть студентов пришла на точные науки исключительно ради этих элементов.

Здесь возникает извечный вопрос: а какой делать диплом? С одной стороны, чтобы были элементы информатики, а с другой – чтобы была польза человечеству в лице научной общественности, которая, несмотря на сопротивление, образованием всё же занимается.

Я рекомендую обратить внимание на возможность доработки и прежде всего объединения уже имеющихся свободных научных программных инструментов. Очевидно, что с нуля поднять серьёзный проект студенту не под силу, но, например, добавить качественную визуализацию, которая есть в проекте *MathGL*, в проект *FriCAS*, где для этого до сих пор используется замечательный, но не всегда эстетичный *gnuplot*, вполне реально. У меня есть некоторое количество идей на тему, куда можно вложиться; возможно, имеет смысл их как-то оформить – пишите письма с вопросами и предложениями.

P.S. В *emacs org-mode* добавили Google Maps. Позитивненько.
E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Сохраняться в облаках 54
Флэшки – вчерашний день. **Знди Ченел** берет Ubuntu One и отправляется в небо, где данные доступны отовсюду.



Держать дела в порядке 58
Завали на работе легко разгребается по принципу «трех гвоздей», но есть и более продвинутые методики, утверждает **Дмитрий Дроздов**.



Чинить файлы 62
Представьте: ваш USB-брелок «посыпался» – что делать? **Бен Мартин** сумеет восстановить файлы с *par2*.



Многомерному анализу 64
Устали от учебных примеров? **Вячеслав Олоничев** и **Андрей Сенов** разберут для вас промышленное OLAP-приложение.



Быстро программировать с Python 68
Надоели подготовительные этапы в начале каждого проекта? **Ник Вейч** и **Quickly** создадут законченную программу на восьми страницах.



Рисовать в Perl 76
Звание «язык для обработки текста» не означает, что в нем нельзя обрабатывать графику. **Михаил Смирнов** представляет *GD*.



Оптимизировать код 80
Программы должны выполняться быстро, даже если они написаны на StarBasic. **Александр Маджугин** объяснит, как этого достичь.



Проверять почту в Mutt 84
Хакеры не запускают X, чтобы проверить почту: у них есть *Mutt*. **Шашанк Шарма** откроет терминал и все объяснит.

Совет месяца: Двойное зрение



Иногда бывает нужно следить за файлом журнала работающей программы. Обычно для этого открывают два окна терминала: одно – для приложения, другое – для команды `tail -f logfile`. Можно использовать вкладки, но тогда в каждый момент времени вы будете видеть что-то одно; или *Screen*, но он разбивает окно по горизонтали, тогда как вам (особенно если у вас широкоформатный монитор) было бы удобнее по вертикали.

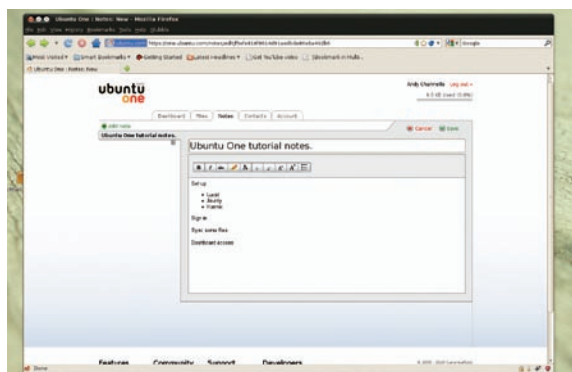
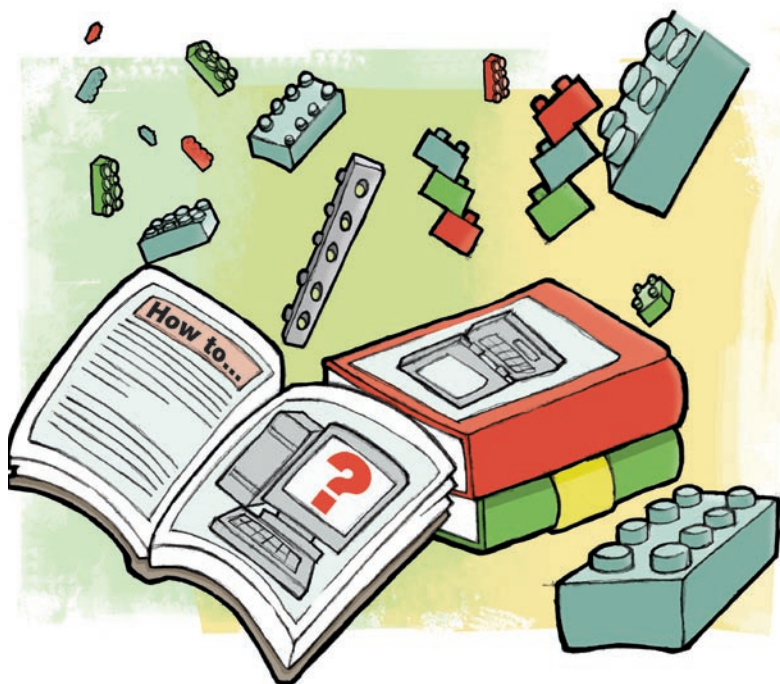
В KDE4, у *Konsole* появился новый режим разделенного просмотра, но на первый взгляд он не так уж полезен – слева и справа (или сверху и снизу) выводится один и тот же терминал. Однако, если у вас открыто несколько вкладок, вы можете распо-

ложить каждую из них в своей области окна. Теперь запускайте в каждой из вкладок свою команду и следите одновременно за двумя вещами: например, самой программой и ее журналом.

Можно разбить окно больше, чем один раз, и отслеживать сразу несколько журнальных файлов, запущенных программ или их произвольную комбинацию. По умолчанию, области имеют одинаковый размер, но активную можно расширить или, наоборот, сузить, так что можно создать себе боковую панель для мониторинга и основное окно для работы с программой. Нам не хватает лишь возможности разбить окно одновременно по вертикали и по горизонтали: ах, разработчики!

Ubuntu. Работаем

Прощай, USB-брелок: будем сохранять файлы онлайн и обращаться к ним из любого места с интернет-доступом. Энди Ченнел расскажет все подробно.



Кроме файлов, со всеми выбранными устройствами можно также синхронизировать заметки и контакты.

ubuntuone-client-gnome. Выберите пакет и нажмите кнопку Применить для его установки.

Возникнет проблема зависимостей, которую необходимо решить. Откройте ваш Терминал (Приложения > Стандартные > Терминал) и введите `sudo apt-get install python-httplib2`. Вам предложат ввести пароль, после чего сработает установка, которая позволит вам запустить программу.

Создание учетной записи

Вы можете запустить сервис, выбрав Система > Настройки > Ubuntu One. Пользователи Lucid Lynx (10.04) также найдут пункт для запуска в меню Me в правом верхнем углу экрана. При первом запуске вам предложат создать учетную запись или войти в существующую. Щелкните на кнопке New Account [Новая учетная запись], введите адрес электронной почты и распознайте текст, доказав, что вы не робот, а затем нажмите «Продолжить». Проверьте свою электронную почту и найдите письмо от Login Service. Там вы найдете ссылку, которая запустит финальную часть процесса установки, когда вам будет предложено ввести свое полное имя и пароль.

После нажатия кнопки Continue [Продолжить] вы войдете в службу и можете добавить компьютер в «сеть». Необходимо вручную добавить каждое устройство, которое вы хотите синхронизировать, следуя инструкциям, приведенным выше, но вместо выбора новой учетной записи, выберите Sign In [Войти в систему] на экране Ubuntu One.

В двух самых последних выпусках Ubuntu вы увидите маленькую иконку в правом верхнем углу экрана, которая показывает, что вы подключены к услуге. Можно нажать на нее, чтобы изменить различные настройки. Сервис добавится в ваш домашний каталог и меню Переходы для быстрого доступа.

Суть идеи в том, что вы перемещаете файл или несколько файлов в папку Ubuntu One, и он будет загружен на облако и доступен на всех машинах с подключением к сети. Если подключенные ма-



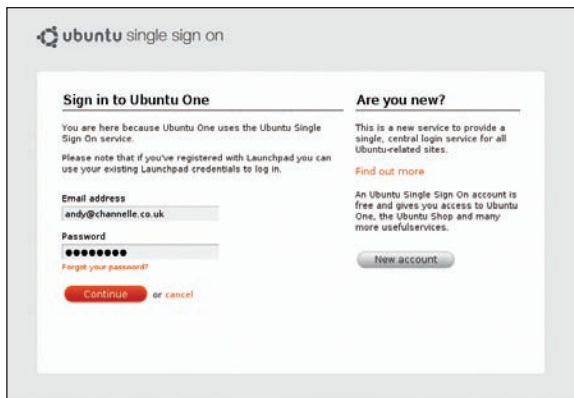
Наш эксперт

Энди Ченнел
Энди, видимо, будет делать Первые шаги в Linux вечно, а технологиями он заинтересовался, открыв для себя Dragon 32.

Для начала работы с Ubuntu One, войдите или создайте новую учетную запись.

Ubuntu положил глаз на работу в облаке, где вы не ограничены физическими возможностями хранилища данных, а ваша информация доступна на любом web-совместимом устройстве. Мы установим облачную службу Ubuntu под названием Ubuntu One, и настроим ее, чтобы получить доступ к файлам, настройкам и документам с нескольких компьютеров. А также добавим на рабочий стол Ubuntu «заоблачный» офисный пакет и многое другое.

Если вы работаете на Ubuntu 9.10 или 10.04, то клиент уже установлен, а значит, вам осталось лишь подписаться на эту услугу. А если у вас более старая версия, запустите Synaptic (Система > Администрирование > Менеджер пакетов Synaptic), затем найдите



Месяц назад Добавили шик в Linux, анимировав презентации OOo Impress.

В облаках

шины выключены, синхронизируемый файл будет скачан при первом запуске. Так вы можете поработать над изображением *GIMP* на своем ноутбуке, сохранить его на диск *UO* и продолжить работу над той же задачей на настольном компьютере.

А если вдруг ваш ноутбук украдут, можно отключить его от своей сети и быть уверенным, что ваша коллекция документов по-прежнему доступна с другого компьютера, или, если у вас только один компьютер, через сайт *Ubuntu One*.

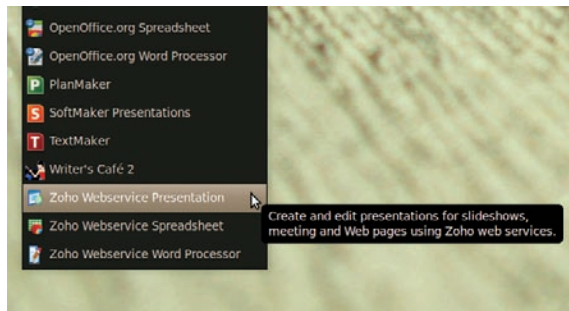
Хотя вы можете управлять файлами через папку *UO*, более продвинутые опции доступны через *web*-сайт службы. Доступ можно получить либо через ссылку *Manage Account* [Управление учетной записью] в *UO Preferences* [Настройках *UO*], либо по ссылке <https://one.ubuntu.com>.

Берем на заметку

Выберите *Notes* [Заметки] и нажмите кнопку *Add Note* [Добавить], чтобы перейти к интерфейсу ввода для вашей заметки. Нажмите *Save* [Сохранить], и заметка будет сохранена в вашем *UO*-пространстве.

Заметки полезнее всего, если вы можете синхронизировать их с вашим рабочим столом. Для этого выберите *Приложения > Стандартные > Заметки Tomboy*. Мы уже рассматривали это простое приложение для работе с заметками раньше, но в своем последнем воплощении его можно синхронизировать с *Ubuntu One* (даже если вы используете *Tomboy* на *Windows* или *OS X*). Чтобы установить это приложение, выберите *Правка > Настройки* и откройте вкладку *Синхронизация*. Выберите *Tomboy Web* из выпадающего меню *Службы* и нажмите *Подключиться*. Откроется браузер на сайте *UO*, и вы сможете «добавить» компьютер к сети. После этого закройте браузер и перейдите в *Tomboy*. Нажмите кнопку *Сохранить*, и ваши заметки будут синхронизированы с онлайн-овой службой. Это двусторонний процесс, поэтому все заметки, которые вы написали в *Tomboy*, будут добавлены в онлайн-овые сервисы и наоборот. Как и с папками, можно синхронизировать несколько экземпляров *Tomboy* с одной учетной записью; таким образом вы всегда будете иметь самые последние заметки, даже если у вас под рукой только смартфон с *web*-браузером.

Хотя *UO* предлагает хранение, существует также расширение для интеграции ряда онлайн-приложений от *Zoho* с вашим стан-



› *Zoho* можно добавить к стандартному меню рабочего стола, для удобного доступа к *web*-приложениям.

дартным рабочим столом. Пакет скорее всего будет доступен по умолчанию в следующей версии *Ubuntu*, но если у вас *10.04*, установите его, открыв терминал и набрав

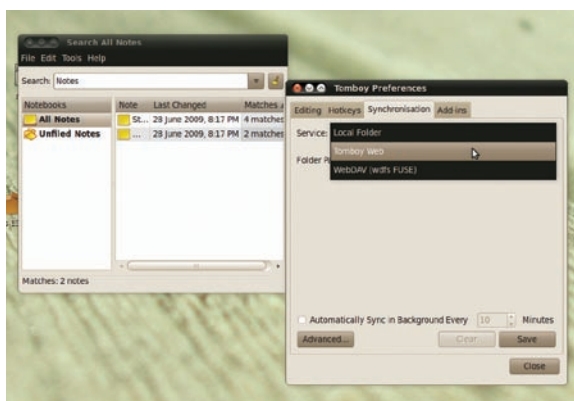
```
sudo apt-get install webservice-office-zoho
```

Введите свой пароль, и пакет будет установлен. Вы найдете три новые иконки в меню *Приложения > Офис*, включая презентации и обработку текстов и электронных таблиц. Вы не вошли в систему, поэтому вы не можете просто сохранить ваши документы на сервере *Zoho*, как при обычном использовании. Вместо этого, выберите опцию *Экспорт* (третья кнопка слева на верхней панели) и формат для сохранения. Чтобы изменить загруженный файл в *Zoho*, щелкните правой кнопкой мыши и выберите *Открыть с помощью > Zoho Webservice 'Application'*.

Если у вас уже есть учетная запись *Zoho*, а браузер по умолчанию – *Firefox*, есть второй способ интеграции службы с вашим рабочим столом. Запустите *Firefox* и перейдите к желаемой службе *Zoho*. Войдя, вы увидите стандартный интерфейс в главном окне браузера. В меню *Tools* [Сервис] выберите команду *Convert Website To Application* [Конвертировать сайт в приложение]. В появившемся диалоговом окне *Prism* можно выбрать, создать ли ярлык на рабочем столе, установить иконку и выбрать элементы браузера, которые нужно показать либо спрятать. Нажмите кнопку *OK*.

Ubuntu не позволит вам запуска по двойному щелчку, потому что это «недоверенная» пиктограмма. Чтобы исправить это, щелкните правой кнопкой мыши на значке, выберите *Свойства*, а в разделе *Разрешения* выберите пункт *Выполнить*.

Преимуществом этого метода является то, что вы не ограничены только *Zoho* – вы можете сохранить и получить доступ к файлам, хранящихся на других облачных сервисах. Например, вам потребуется всего несколько минут, чтобы правратить *Google Docs* (файловый менеджер или отдельные компоненты), *Dropbox* и *Last.fm* в *web*-приложения. Недостатком является то, что этот метод не позволяет сохранить локальную копию файлов, хотя и *Google*, и *Zoho* можно настроить на автономный доступ.



› Заметки на рабочем столе обрабатываются *Tomboy*, а затем синхронизируются с облаком; таким образом, они доступны отовсюду.

Облачная головная боль

Готовя этот материал, мы столкнулись с одним из главных разочарований использования облаков: ограниченные или нестабильные сетевые соединения, которые могут реально затормозить систему. *Ubuntu One* предлагает преимущества как локального хранения, так и на облаке, что позволяет работать с документами на рабочем столе, а затем синхронизировать их с облаком, если и когда оно будет доступно.

KDE: Создаем

Ускорьте свою работу, создав себе комбинации клавиш для повседневных задач.

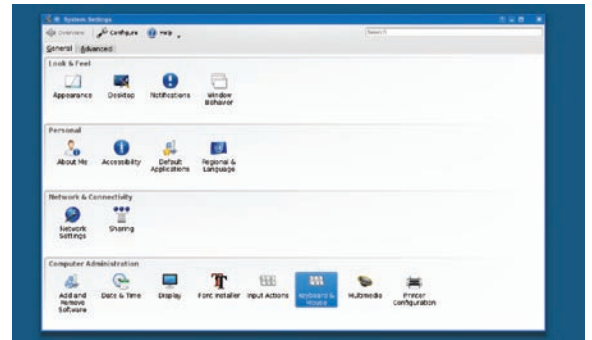
Почти у каждого пользователя компьютера настает момент, когда мышь начинает мешать. Это начинается тогда, когда вы обнаружите, что почти во всех приложениях Ctrl+C и Ctrl+V будут копировать и вставлять любые выбранные элементы. Затем вы замечаете, что нажать Ctrl+S гораздо быстрее, чем гонять курсор по Файл > Сохранить, и хотя это экономит от силы секунду или две, лишние секунды копящаяся, особенно если вы параноик и норовите сохранять работу каждые пару минут.

Linux, как вы, наверное, уже знаете, почти бесконечен с точки зрения настроек, тем более по части клавиатуры и мыши. На этом уроке мы займемся сочетаниями клавиш и сделаем так, чтобы большинство ваших наиболее частых задач выполнялось без отвода рук от клавиатуры. А в следующий раз мы рассмотрим другие методы ввода.

Управление с клавиатуры возможно практически для всего; открыв свое любимое приложение, вы увидите сочетания клавиш, указанные рядом с пунктами меню. Существуют стандартные привязки, такие как Ctrl+O для открытия, Ctrl+P для печати и Ctrl+A для выбора всего. Они приняты в большинстве приложений, а кроме них, есть еще специальные сочетания в отдельных программах, такие как Ctrl+m в *Konqueror*, который включает или отключает меню, или Пробел, позволяющий выбрать фото, если вы работаете в *Cheese*.

Как следствие, при невнимательности довольно легко насоздавать конфликтующие сочетания клавиш; поэтому мы будем назначать задачам комбинации клавиш наподобие Ctrl+F6 или Alt+Shift+2. Хотя отсутствие конфликтов не гарантируется, мы, по крайней мере, избежим их в рамках стандартных программ KDE, используя Keyboard Shortcut Manager этого рабочего стола. В пошаговой инструкции на соседней странице, мы также заостримся на некоторых изменениях, которые можно внести в обычные приложения Linux.

Начните с открытия меню KDE и выберите Компьютер > Параметры системы. В разделе Администрирование есть две записи со значками клавиатуры, просто чтобы сбить вас с толку. Пока



Экран настроек Клавиатура и мышь позволит назначить свои комбинации клавиш для приложений KDE.

проигнорируем элемент Параметры ввода и выберем Клавиатура и мышь. В меню слева выберите Стандартные сочетания клавиш – вы увидите длинный список доступных функций и связанных с ними комбинаций. Давайте изменим первую запись, Фактический размер, пока не связанную ни с какой клавиатурной комбинацией. Выберите ее, чтобы получить доступ к параметрам. Теперь нажмите кнопку Дополнительно и введите сочетание клавиш, которое вы хотите ассоциировать с этим действием; в нашем случае мы выбрали Ctrl+0 (это ноль, а не буква O, почти везде используемая для открытия), и, наконец, нажмите на название действия еще раз.

Выберите свои клавиши

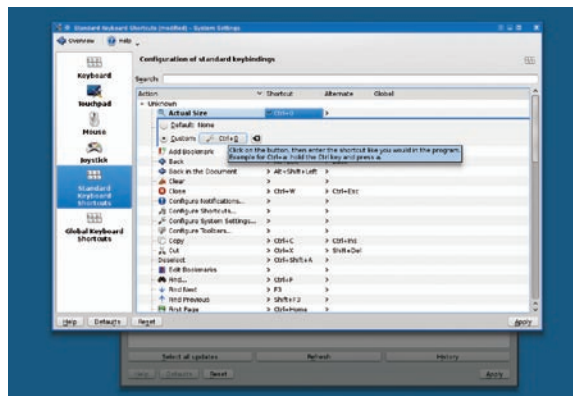
Определяя сочетания, старайтесь мыслить логически. Например, если вы установили для Увеличить клавиши Ctrl+[+] (клавиша плюс), было бы умно для Уменьшить назначить Ctrl+[-]. Если вы заодно установите Ctrl+9 для Увеличить по ширине страницы, у вас образуется кластер сочетаний клавиш для масштабирования изображений в определенной зоне клавиатуры. Также можно для мнемоники брать первую букву команды. Ctrl+t, например, может показывать или скрывать панель инструментов [Tools], хотя кое-где она уже открывает новую вкладку.

Завершив настройки, нажмите кнопку Применить в нижней правой части окна, чтобы сохранить изменения. Вы увидите уведомление, предупреждающее, что изменение настроек может привести к конфликту с некоторыми приложениями, и работающие программы нужно будет перезапустить, чтобы изменения вошли в силу. Нажимаем ОК, и все готово. Проверьте меню, и вы увидите новые сочетания клавиш к соответствующим пунктам.

Эти настройки применяются к интегрированным в KDE приложениям, в том числе *KOffice*, *Konqueror* и *Dolphin*, но не работают для других приложений, таких как *OpenOffice.org* или *GIMP*. Эти популярные программы имеют собственные настройки, которые мы рассмотрим подробно в пошаговой инструкции.

Скорая помощь

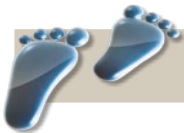
Согласованность между приложениями – дело хорошее: использование клавиатуры становится значительно интуитивнее. Только по этой причине и стоит менять настройки комбинаций клавиш в приложениях.



Установка собственных комбинаций сэкономит вам массу времени.

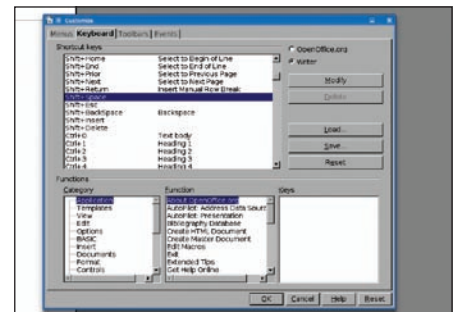
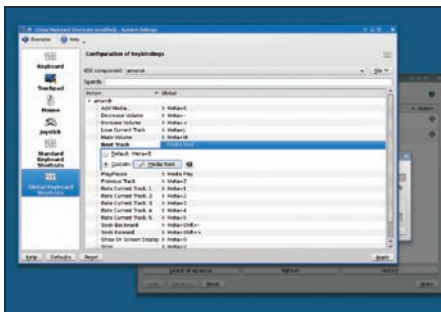
» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

ГОРЯЧИЕ КЛЮЧИ



Шаг за шагом: Настройка других приложений

Если ваши любимые программы не состоят в рядах KDE, для них все равно можно добавить свои сочетания клавиш. Следуйте инструкциям.



1 Используйте медиа-клавиши

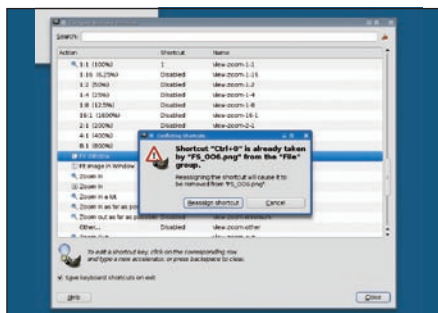
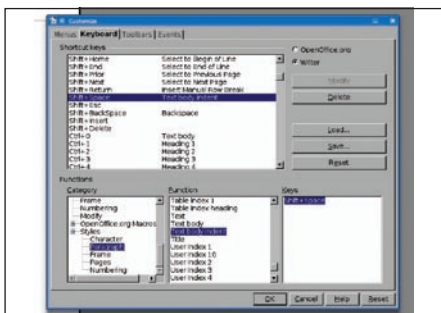
У большинства клавиатур в верхней части есть ряд так называемых медиа-клавиш, для управления воспроизведением фильмов или музыки. Чтобы изменить их настройки в KDE, перейдите в Клавиатура и мышь > Глобальные горячие клавиши и выберите *Аналог* в списке компонентов KDE. Определите сочетания клавиш, как делали раньше. Нажмите Дополнительно, затем кнопку Определить, и укажите подходящее сочетание клавиш.

2 Кнопка Windows

Немало клавиатур имеют клавишу Windows рядом с кнопкой Alt. KDE считает, что это мета-клавиша, которую можно использовать так же, как Alt, Ctrl или Shift. Это, несомненно, открывает намного больше возможностей для настройки, поскольку большинство приложений ее не используют. Он может быть настроен как указано выше.

3 OpenOffice.org

Кроме стандартных ключей, *OpenOffice.org* имеет массу уже установленных сочетаний клавиш. Чтобы определить свои собственные, откройте любое приложение и выберите Сервис > Настройка. Загляните на вкладку Клавиатура для просмотра имеющихся вариантов. В верхней части окна вы увидите, какие комбинации уже используются. Выбирайте доступные сочетания клавиш, такие как Shift+Пробел.



4 Определите действие

Теперь используйте три колонки выбора в нижней части окна, чтобы определить действия. На этом экранном снимке мы используем Shift+Пробел для доступа к Стилю > Абзац > Текст с отступом, то есть можем выбрать фрагмент текста, а затем применить соответствующий стиль клавишами, вместо того, чтобы углубляться в дебри меню стилей.

5 Команды GIMP

Вы сможете настроить сочетания клавиш *GIMP* по Правка > Сочетания клавиш. Вы увидите, что по умолчанию для большинства инструментов горячие клавиши отключены, так как обычно дизайнеры предпочитают мышь. Тем не менее, можно настроить любой из вариантов через меню. Отметьте тот, который вы хотите, выберите Новый акселератор и введите новую комбинацию клавиш. *GIMP* предупредит вас, если возникнет конфликт.

6 Управление Firefox

Изменить сочетания клавиш в *Firefox* не так-то просто, и лучшим выходом, возможно, будет использовать *Mouseless* – расширение Рудольфа Ноз [Rudolf Noe]. Оно позволяет просматривать Сеть с помощью клавиатуры, добавив число (выбираемое при нажатой клавише Ctrl) к каждой ссылке на странице. Это требует привычки, но пригодится тем, кому мучительно покидать клавиатуру. [LXF](#)

» Через месяц Синхронизируем данные с телефонами Android и поиграем с мышью.

GTG: Дела

Завалили работой, и вы не знаете, с чего начать? Применив идеи Дэвида Аллена под руководством **Дмитрия Дроздова**, вы сделаете собственный отпуск ближе.



Наш эксперт

Дмитрий Дроздов

Упорно использует Linux в офисной работе и не собирается отказываться от удобной и надежной среды.

Дэвид Аллен [David Allen], автор книги «Getting Things Done» (в русском переводе – «Как привести дела в порядок»), изложил набор весьма полезных советов по повышению производительности труда. В двух словах, основная идея состоит в том, что пока вы работаете над какой-то задачей, на все новые дела вы вообще не отвлекаетесь. Как только наступит перерыв в текущей работе и настанет момент для сортировки накопившихся дел, каждое из них либо заносится в один из списков дел для последующего выполнения, либо вообще отбрасывается («мусор»), либо, если на него не требуется много времени, выполняется немедленно. При этом списков задач может быть много; более того, одна и та же задача может оказаться в разных списках. Например, задача «подготовить иллюстрации для статьи о GTG», очевидно, может быть решена только на том компьютере, где установлен GTG (назовем этот список «компьютер с Linux»), но при этом она может быть, скажем, в списке «обязательства перед третьими лицами».

Впрочем, пересказать даже основные идеи упомянутой книги парой предложений нельзя. Однако популярность изложенного подхода привела к появлению приложения, которое помогает в реализации основных идей Аллена. Имя ему – *Getting Things GNOME!* (GTG), что явно указывает на «родной» рабочий стол программы. В названии очевидна игра слов и букв, но GTG примечателен не только этим.

GTG и все-все

Сразу же возникает вопрос: «А чем GTG отличается от разных менеджеров задач?» В первую очередь – тем, что здесь созданы действительно удобные средства для формирования списков задач и подзадач, их просмотра, редактирования и связи с другими

инструментами организации работы. Уже этого достаточно, чтобы попытаться попробовать этот инструмент.

В GTG не просто ведётся список задач, что делают все приложения такого рода, а достаточно полно реализуется концепция Аллена по наведению порядка в делах. Среди опробованных автором списков задач весьма полная реализация такого подхода встретила именно в GTG. Правда, чтобы оценить это, надо достаточно активно использовать в работе идеи из первоисточника. Но даже если и не быть сторонником организации труда по Аллену, то GTG будет весьма полезен в повседневной практике.

Функциональность GTG может достаточно просто наращиваться путем подключения дополнительных модулей. Здесь очень удачно реализован механизм подзадач и практически любых меток для задач и подзадач. GTG дает единое место хранения и самого списка задач, и заметок, имеющих к ним отношение. GTG достаточно хорошо продуман для удобства работы со всеми этими вещами. Проект активно развивается, и вы даже можете предложить авторам внести какую-то дополнительную функциональность.

Установка GTG

GTG, скорее всего, имеется в репозитории вашего дистрибутива. Несмотря на это, устанавливать его лучше с сайта разработчиков: проект активно развивается, и установка последней версии – разумное решение. На момент написания статьи вышла версия 0.2.4, которую можно скачать с <http://launchpad.net/gtg/0.2/0.2.4>.

GTG написан на Python (версия не ниже 2.5). Среди зависимостей – PyGTK, Python ConfigObj, Python GObject, Python XDG. Впрочем, для модулей расширения могут потребоваться и другие библиотеки Python (но об этом позже).

GTG можно запустить и без установки на компьютере. Для этого достаточно распаковать скачанный архив командой `tar xzvf gtg_версия.tar.gz` в какой-нибудь директории и выполнить скрипт `gtg`. Однако, если программа окажется полезной, в этом же каталоге можно дать команду (от имени root) `python setup.py install` (в современных системах на основе Debian к этой команде требуется добавить ключ `--install-layout=deb`). После этого можно будет запускать GTG из системного меню или командой `gtg`.

GTG хранит все данные в файлах формата XML, которые располагаются в директории `~/local/share/gtg`. Это позволяет при необходимости легко перенести свой список задач на другой компьютер. Ознакомившись с содержимым этого каталога, несложно установить, что файл `projects.xml` содержит всего лишь ссылку на XML-файл с активным списком задач, и если автоматически сгенерированное имя вашего списка задач вас не устраивает, то можно немного поправить файл и изменить это. Два остальных XML-файла, очевидно, содержат списки ваших задач и меток. В этой же директории сохраняются несколько резервных копий, что может спасти данные, если со списком задач произошли какие-то неприятности (например, вы что-то ошибочно удалили).

Сразу после первого запуска GTG выведет на экран ваши новые задачи. На самом деле это довольно исчерпывающее опи-



В порядке

сание программы. Двойной щелчок на любом заголовке задачи открывает её описание в отдельном окне. Вообще, все советы из начального списка стоят того, чтобы прочитать их достаточно внимательно, благо есть переводы на многие языки, включая русский.

Изучаем возможности

Самая замечательная функция *GTG* – пометка задач. Эта возможность позволяет легко сортировать задачи по любому разумному критерию. Например, над некоторыми задачами можно работать либо только в офисе, либо только дома. Первые, очевидно, лучше всего связать с меткой *офис*, а вторые – *дом*.

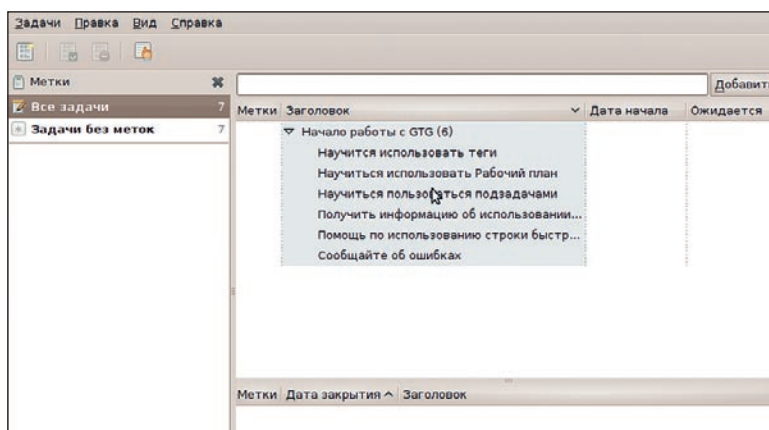
Чтобы пометить задачу, достаточно предварить символом @ любое слово в её описании. После ввода такое слово будет автоматически выделено жёлтым фоном. Например, @офис или @дом пометит задачи для выполнения в офисе или дома. Число меток для каждой задачи не ограничено, лишь бы было удобно. По практическим соображениям список меток удобнее всего размещать в одной строке сразу после заголовка задачи. При редактировании описания задачи *GTG* иногда самостоятельно вставляет между метками запятые: они никак не влияют на работу, не обращайте на них внимания. Опять-таки по практическим соображениям метки не должны быть длинными.

Весь список меток доступен в левой части окна (его можно включить или выключить командой меню Вид > Панель меток или клавишей F9). Список формируется динамически (т.е. по мере создания меток в описаниях задач) и сортируется по алфавиту.

В *GTG* можно создать нужную иерархию меток! Для этого достаточно перетащить одну метку мышью на другую, и первая сразу же станет подчинена второй. Для возврата подчиненной метки на верхний уровень достаточно «вытащить» её в конец списка.

Если щелкнуть правой кнопкой мыши по метке, то можно задать цвет фона, на котором будут выводиться задачи с ней. Эта возможность напоминает цветные маркеры для выделения разных типов задач. Если задача имеет несколько меток, то все они будут отображены цветными квадратиками перед описанием задачи в списке. Щёлкнув мышью по какой-то метке в панели, можно отобразить в списке только те задачи, которые имеют данную метку.

Поскольку *GTG* создавалась программистами с опытом работы в командной строке, много действий можно выполнять и без использования мыши. Про @ мы уже говорили; дефис в начале строки добавляет в описание подзадачу – она появится автоматически после завершения ввода. Название подзадачи станет ссылкой на её описание. Подзадаче будут автоматически добавлены все метки, которые были у родительской задачи на момент создания. Таким образом можно создавать любую иерархию задач для выполнения большого задания.



Для оперативного создания задачи без подробного описания удобно пользоваться строкой быстрого ввода, которая располагается над списком задач. Например, ввод в этой строке «Проверить работу Apache due:0515 tags:дом» приведет к появлению в списке соответствующей задачи, помеченной меткой *дом* и датой выполнения до 15 мая. В строке ввода можно использовать ключевые слова (после них обязательно двоеточие): *tags;*, *due;*, *defer;*. За первым ключевым словом можно перечислить через запятую все необходимые метки. *Due* – дата, до истечения которой должна быть выполнена задача. *Defer* – срок, на который нужно отложить начало выполнения задачи. Даты могут задаваться в нескольких форматах: *yyyy-mm-dd*, *yyyymmdd* или *mmdd* (0515, в этом случае используется текущий год), либо можно задать описания (работают и по-русски) «сегодня» (*today*), «завтра» (*tomorrow*) или день недели (будет подставлена дата ближайшего дня недели).
Вообще, возможность задавать даты в наиболее естественных для обычного человека терминах – сделать *завтра* или *на следующей неделе* – еще одно большое удобство *GTG*. Эта возможность прослеживается везде, даже в отображении срока выполнения задач.

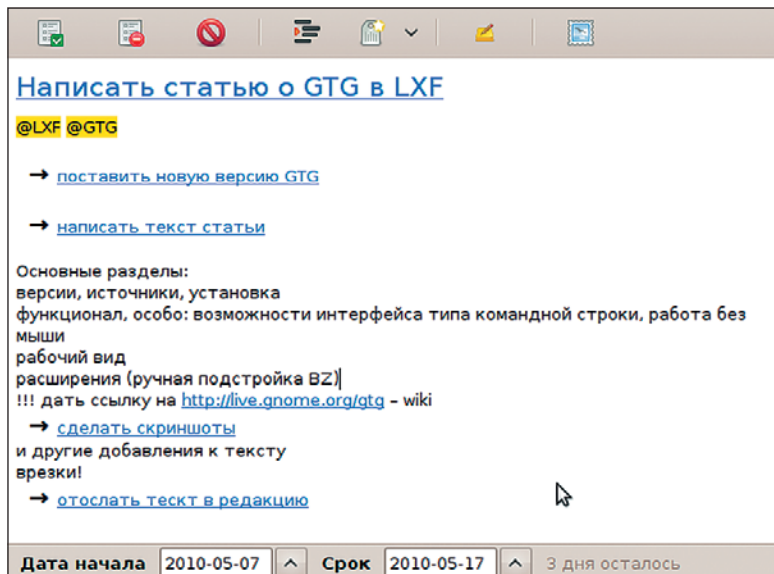
Если в момент создания новой задачи в списке меток было что-то выбрано, то вновь создаваемой задаче будет автоматически присвоена эта метка (если метки заданы явно при помощи ключевого слова *tags;*, то выбранная метка будет автоматически добавлена к перечисленным).

В окне описания задачи можно разместить любой текст, который может содержать ссылки на страницы Сети. При щелчке по URL-ссылке будет автоматически запущен браузер, установленный в системе по умолчанию. Это удобно для ведения различных заметок, которые необходимы для выполнения задания. При этом текст, метки, подзадачи могут как угодно чередоваться в описании

» Окно *GTG* при первом старте. Каждая задача – весьма подробная справка по работе с программой.

«Много действий можно выполнять и без использования мыши.»

» Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.



В описании задачи могут чередоваться метки, подзадачи, текст и даже «кликабельные» интернет-ссылки.

задачи. Это позволяет чрезвычайно гибко использовать GTG, хотя может и привести к немалому беспорядку.

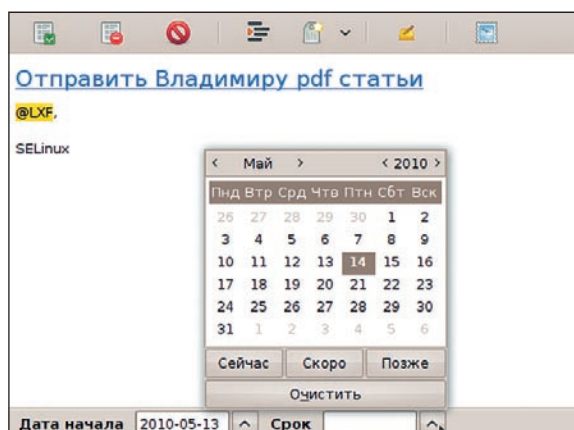
Рабочий план

Чтобы выполнить какую-то работу, полезно видеть лишь те задачи, над которыми можно трудиться в данный момент времени. Такой список в GTG называется «Рабочий план». Для его вывода надо вызвать команду меню Вид > Рабочий план или воспользоваться кнопкой панели инструментов. Чтобы задача была отражена в рабочем плане, нужно соблюсти два условия:

- » Дата начала выполнения задачи должна уже наступить.
- » Все подзадачи этой задачи уже должны быть выполнены. Несмотря на парадоксальность последнего утверждения (логика почему-то шепчет: «все задачи завершены – значит, и всё дело тоже сделано»), это очень удобный механизм для контроля именно *полного завершения* дела из нескольких шагов. Действительно, на момент выполнения последней подзадачи ситуация со всем делом в целом могла поменяться, и GTG предлагает, прежде чем щёлкнуть по кнопке Завершено, на мгновение задуматься и пробежаться глазами по описанию задачи: возможно, что-то придется и отредактировать, добавить.

Все другие задачи в рабочем плане не отображаются, ведь над ними нельзя работать в текущий момент времени: либо для их выполнения что-то очевидно не готово, либо еще не пришла пора.

Сроки начала работы и выполнения задач можно задавать и датами, и более естественными понятиями: сегодня, завтра, ...



Кроме этого, в рабочий план не попадут задачи с метками, у которых сброшен признак Показывать в рабочем плане. Признак устанавливается (точнее, сбрасывается, поскольку по умолчанию он установлен) для каждой метки при редактировании её свойств из контекстного меню панели меток. Так можно легко вести список дел, под названием «Может быть, когда-нибудь» (естественно, само название метки для удобства ввода должно быть существенно короче, например, @maybe).

Рабочий план может быть отсортирован по любой из меток. Увидеть, что надо сделать дома или на работе, можно буквально одним щелчком мыши, если метки расставлены разумно!

Расширения

Несмотря на молодость проекта, GTG уже обзавелся расширениями. Установить их можно командой меню Правка > Параметры. На вкладке Модули отмечаются те из них, что необходимы вам в работе. Если модуль выделен серым шрифтом, это значит, что для данного расширения не хватает каких-то библиотек; их надо предварительно установить. Впрочем, узнать, каких именно библиотек не хватает, легко: достаточно щёлкнуть по кнопке О модуле. После установки всех нужных библиотек можно включить понравившийся модуль.

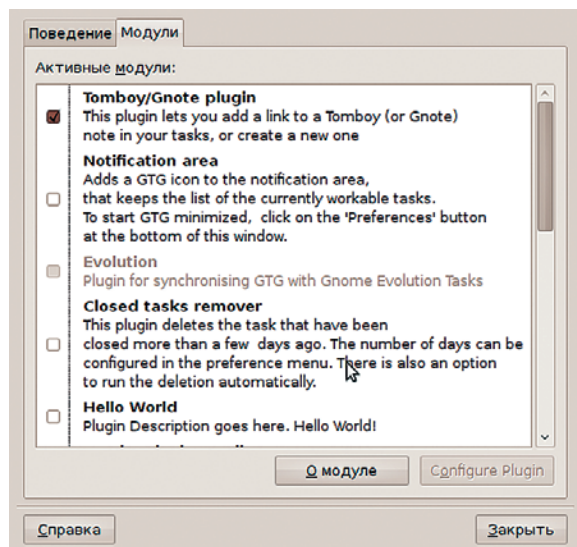
Среди наиболее полезных расширений – интеграция с электронной почтой, заметками Tomboy, онлайн-сервисом напоминаний Remember The Milk (www.rememberthemilk.com). Для использования последней возможности надо иметь учетную запись на этом сервисе; всё остальное сделает GTG. Правда, разработчики специально указывают, что поскольку GTG не имеет официального одобрения от RTM, работоспособность не гарантируется (однако, как показывает опыт, все работает вполне нормально). Кстати, сервис RTM имеет и русский вариант интерфейса.

Посылка описания задачи электронной почтой позволяет весьма эффективно организовать командную работу. После включения расширения в инструментальной панели редактора задачи появляется кнопка, позволяющая буквально одним щелчком подготовить письмо, заголовок которого будет соответствовать названию задачи, а в теле письма будет всё её описание, включая метки. (Ещё раз оцените удобство держать название задачи и всякие заметки в компактной форме!) Остается только выбрать адресатов, возможно, снабдить письмо какими-то дополнительными распоряжениями или просьбами, и отправить.

Среди тестируемых сейчас расширений также заслуживает упоминания модуль взаимодействия с системой Bugzilla. Пока он обеспечивает работу «из коробки» лишь с Bugzilla проектов GNOME, Mozilla и Freedesktop.org. Однако, если немного поработать с исходным кодом модуля, удастся обеспечить приемлемую

Действие	Комбинация клавиш
Отметить задачу как выполненную	Ctrl+d
Отменить задачу	Ctrl+i
Новая задача	Ctrl+n
Новая подзадача к выбранной задаче	Ctrl+Shift+n
Выход из GTG	Ctrl+q
Добавить метку к выбранной задаче	Ctrl+t
Отменить действие в редакторе задачи	Ctrl+z
Повторить действие в редакторе	Ctrl+y
Вывести/скрыть панель меток	F9
Вывести/скрыть список выполненных задач	Ctrl+F9
Удалить задачу	DEL
Начать редактирование выбранной задачи	ENTER
Завершить редактирование выбранной задачи	ESC

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



» Вкладка «Модули»: тут включаются и отключаются расширения.

работу и с другими, например, локальными системами отслеживания ошибок.

Повышаем производительность

В *GTG* есть много полезных возможностей, которые повышают производительность работы с приложением. О средствах ввода с клавиатуры уже упоминалось. Естественным дополнением к ним являются комбинации клавиш. Основные приведены в таблице.

Есть один момент, который поначалу может буквально поставить в тупик: в *GTG* нет понятия приоритета задач. Не будем спорить, хорошо ли это – имея развитый механизм иерархических меток, можно создать даже нечто большее, чем просто приоритеты. Чтобы получить аналог приоритетов, просто создадим метки @1, @2 и @3. Самые важные задания будем отмечать @1, а самые незначительные – @3 или вообще оставим без меток. В панели меток мышью создадим такую иерархию:

```
@3
-@2
--@1
```

Тогда, выбирая задачи, отмеченные @3, мы увидим в списке и задачи приоритетов 1 и 2, и т.д. Ну чем не привычная по другим спискам задач система приоритетов? А как вам такая идея: добавить к цифрам букву о или h?

Разные приёмы эффективного использования *GTG* можно найти на странице «Советы и трюки» по адресу: <http://gtg.fritalk.com/post/2010/02/03/GTG-tips-and-tricks2> (на английском языке).

Работаем эффективно

Наконец, поделюсь несколькими приёмами работы с *GTG*, которые заметно упрощают ведение списка дел. Конечно, на этот счет может быть множество мнений, и я несколько не претендую на истину в последней инстанции.

GTG (да и другие менеджеры задач) принципиально не подходит для ведения списка дел, время выполнения которых должно быть задано с точностью до минут. Поэтому *GTG* и удобный электронный календарь должны работать вместе. В последнем отмечается всё, что точно «привязывается» к определенному времени: совещания, визиты, поздравления с днями рождения и т.п. Если для какой-то задачи точное время выполнения не задано, тогда ей самое место в *GTG*.

Третьим инструментом организации списка дел является какое-то временное хранилище записей, используемое в периоды, когда *GTG* и календарь недоступны. Это может быть и текстовый

Библиография

Аллен Д. Как привести дела в порядок: искусство продуктивности без стресса. Пер. с англ. – М.: ООО «И.Д. Вильямс», 2008. – 368 с.

файл, и посланный самому себе «мэйл-напоминалка», и просто блокнот с карандашом. Главное – выработать привычку переносить задачи из временного хранилища либо в *GTG*, либо в календарь.


Лучше всего, если задачи заносятся в *GTG* (или календарь, если дело простое и время его выполнения известно) сразу, как только они возникли. Например, вы что-то пообещали во время разговора по телефону – сразу запишите. Для ускорения процесса можно поначалу не присваивать никаких меток.

По мере накопления задач без меток (их легко вывести одним списком) их надо просмотреть; возможно, уточнить формулировки одних, другим – установить подзадачи и сроки. Но главное во время этой сортировки – определить всем задачам метки. На этом этапе часто бывает, что какие-то задачи перемещаются в календарь, а некоторые уже можно не выполнять из-за потери актуальности. По опыту автора, такую операцию достаточно делать ежедневно, и занимает она минут 10–15.

Выработка собственной удобной системы меток – сама по себе важная задача (не занести ли её в *GTG*?). Тут могут использоваться разные подходы: по важности, по месту или условиям выполнения, про проектам. Чтобы учесть опыт других, стоит познакомиться с книгой Аллена, да и со статьями в Интернете. Последних становится, кстати, все больше. Но выработать *удобную для себя* систему меток придётся *самостоятельно*. Благо, динамическое формирование списка меток в *GTG* немало способствовало различным экспериментам, и такая система быстро вырабатывается вроде бы как сама собой.

А теперь представим себе, что у вас есть 15 минут до очередного совещания и вы к нему готовы. Вы в офисе, но при этом относительно свободны, а желания попить кофе нет. Чем занять это время? Нет ничего проще: пара кликов мышью – и вы видите, кому вам нужно позвонить (метка @tlf) сегодня или в ближайшее время. И четверть часа проходит в коротких телефонных звонках, о которых вы бы, возможно, в ином случае просто не вспомнили. А если вы не дозвонились кому-то из этого списка, беда нет, поскольку срок выполнения задания еще не наступил. Так *GTG* (и здравый смысл вкупе с самоорганизацией) помогают разгрузить внимание и добиться высокой производительности.

Getting Things GNOME! в любом случае заслуживает того, чтобы его попробовать, даже если вы не собираетесь использовать в полном объеме концепцию повышения производительности труда Д. Аллена и не были знакомы с ней – потраченное время того стоит. А может случиться и так, что поработав с *GTG*, вы пожелаете побольше узнать о концепциях, которые были положены в его основу, и ознакомиться с книгой.

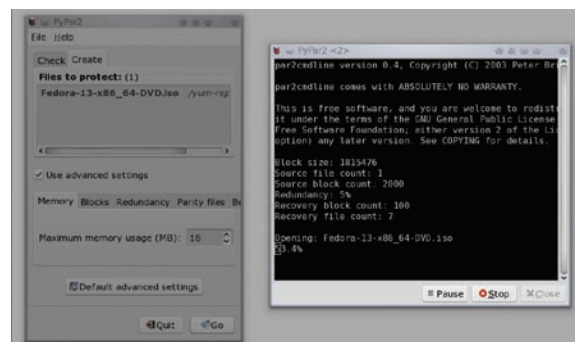
Проект довольно активно развивается; не исключено, что к выходу статьи появится уже версия ветки 0.3, которая обещает быть еще более функциональной и удобной в работе. Порядок в делах нам гарантирован! 

Полезные ссылки

- » <http://gtg.fritalk.com/> – официальная страница проекта *GTG*
- » <https://launchpad.net/gtg> – страница разработчиков проекта *GTG* на Launchpad: источник новых версий, отслеживание ошибок, обсуждение новых функций, работа команды переводчиков
- » <http://live.gnome.org/gtg> – вики проекта (на английском)
- www.rememberthemilk.com – сервис напоминаний Remember The Milk

par2: Проверь и

Бен Мартин расскажет, как с помощью *par2* собрать маленькую армию восстановительных файлов, которая не даст пропасть вашим данным.



» Нажав кнопку Go, вы увидите окно с отображением процесса непосредственно из *par2cmdline*.

Существует три основных режима работы *par2*: создание, проверка и восстановление. Пакет *par2* на машине Fedora Linux называется **par2cmdline** и состоит из команды *par2* и трёх команд-псевдонимов: **par2create**, **par2verify** и **par2repair**. Вызов любой команды оформляется одинаково, тут не запутаешься: **par2create par2-file file1 file2** и т. д., где **file1 file2** и им подобные — имена файлов для создания, проверки или восстановления.

За дело...

Одной командой **par2create** можно сгенерировать восстановительные файлы аж для 32000 файлов. Программа строит пирамиду своих файлов, присваивая им имена с префиксом **par2-file**. Файлы в составе пирамиды прогрессивно увеличиваются в объёме. Если повреждение незначительно, то небольшого файла окажется достаточно, чтобы спасти положение. Катастрофическую потерю данных может залечить только крупный восстановительный файл.

Для демонстрации я взял файл **Fedora-12-x86_64 DVD.iso**. Его объём около 3,3 Гб, а контрольная сумма *md5sum* выглядит как **6dd31e292cc2eb1140544e9b1ba61c56**. Тестирование выполнялось на машине с процессором 2,4 ГГц Intel Q6600 и массивом RAID. Чтобы создать MD5, понадобилось 39 секунд, а *par2* со стандартными параметрами работала около восьми минут. Для создания восстановительных файлов ISO-образа я использовал команду **par2create** в следующей форме:

```
par2create Fedora12Par2 Fedora-12-x86_64-DVD.iso
```

Основной параметр командной строки, интересующий нас при создании файлов *par2* — это **-r**. По умолчанию выставляется **-r5**, для создания избыточности в 5%. Этот параметр определяет размер восстановительных файлов и количество ошибок, которые с их помощью можно исправить.

Время, затрачиваемое на создание файлов, зависит от количества генерируемых данных. Например, если вы хотите получить

Наш эксперт

Бен Мартин

работает с файловыми системами свыше 10 лет. После защиты диссертации он занимается консультированием по вопросам *libferris*, файловых систем, поисковым системам и программированию *Qt/C++*.

Многим известно, что контрольные суммы CRC32, SHA-1 и MD5 позволяют с высокой степенью вероятности судить о сохранности крупных файлов. С помощью *par2* можно определить количество кода для коррекции ошибок, а при необходимости пойти дальше и рендимировать слегка подпорченные файлы. Кто знает: может быть, визит *par2* на старый флэш-накопитель или второй DVD с резервной копией диссертации однажды спасёт ваши труды за 10 лет!

Элегантность программы в том, что можно заранее создать для архива «дерево» корректирующих файлов, и в случае порчи архива достаточно будет только загрузить такое количество «ремонтных» данных, которое соответствует уровню повреждения, чтобы архив был починен. То есть, если в 5-гигабайтный Live-DVD затесалась парочка ошибок, можно загрузить всего один, очень маленький файл для восстановления исправной версии.

Как повысить скорость

Количество используемой памяти определяется параметром **-m**. Стандартное значение **-m16** позволяет использовать только 16 Мб ОЗУ, что для современных машин неактуально. Например, имея 8 Гб ОЗУ, я использую значение **-m512**, и вместо восьми минут нужное количество восстановительных файлов генерируется за семь.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

ПОЧИНИ ФАЙЛЫ

избыточность 10%, то времени на это уйдёт вдвое больше, чем на файлы с 5%-й добавкой.

Восстановительные файлы занимают место и требуют времени для генерации, поэтому неплохо бы выяснить оптимальный объём избыточности. Весьма полезная функция *par2* – выработать дополнительный код для восстановления, который можно использовать в сочетании с готовыми файлами *par2*.

Например, изначально вы сгенерировали файлы *par2* с избыточностью 5%, но вам сообщили о 7%-м повреждении. В этом случае с параметром **-r5 -fNNN** можно сгенерировать 5%-ю добавку, начиная с блока **NNN**. Блок **NNN** должен быть следующим за имеющимися блоками. Чтобы найти **NNN**, найдите наибольший файл *par2* – скажем, **F12.vol063+37.par2**. В нем содержится 37 блоков, начиная с блока 63. То есть изначально было создано 100 блоков, и можно создать дополнительный 5%-й код для восстановления, начиная со следующего блока с параметрами **-r5 -f101** при запуске **par2create**.

Найти и починить

Чтобы узнать, не изменился ли DVD-образ со времени генерации файлов *par2*, скомандуйте:

```
par2verify Fedora12Par2.par2 Fedora-12-x86_64-DVD.iso
```

Если всё в порядке, выведется строка 'All files are correct, repair is not required [Файлы в порядке, исправлений не требуется]'. Задача *par2* – восстановление повреждённых файлов, поэтому я намеренно нарушил часть ISO-образа и восстановил оригинал. Для вставки прямо в файл случайного набора байтов, с заданным смещением от начала файла, можно использовать команду **dd**. А чтобы **dd** не обрезала файл до нуля байт, необходимо добавить ключи **seek** и **conv=notrunc**. Команды **bs** (размер блока) и **count** устанавливают начальную позицию и объём копирования. Следующим набором команд мы введем 4 КБ случайных данных в ISO-образ, начиная со смещения 16 КБ от начала файла.

```
dd \
if=/dev/urandom \
of=Fedora-12-x86_64-DVD.iso \
bs=1 count=4096 seek=16384 \
conv=notrunc
```

После разрушения 4 КБ данных команда **par2verify** должна сообщить о повреждении и количестве блоков, необходимым для его устранения.

```
par2repair Fedora12Par2.par2 Fedora-12-x86_64-DVD.iso
```

```
...
Repair is required.[Необходимо исправление]
1 file(s) exist but are damaged. [1 файл(ы) существует,
но поврежден]
You have 1999 out of 2000 data blocks available.[Вам доступно
1999 из 2000 блоков]
You have 100 recovery blocks available.[У вас есть
100 восстановительных блоков]
Repair is possible.[Восстановление возможно]
You have an excess of 99 recovery blocks.[У вас 99 избыточных
восстановительных блоков]
```

Интернет-стратегия

Восстанавливая файл, полученный из Интернета, лучше начать с загрузки файла *par2* и наименьшего по объёму **volXX.YY.par2**. Если ошибок немного, то **par2repair** тут же исправит их; в противном случае вы узнаете,

сколько дополнительных блоков необходимо для восстановления.

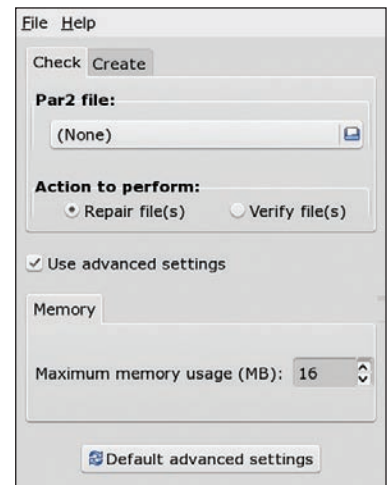
Вооружившись этим знанием, загружайте столько дополнительных файлов **vol.par2**, сколько нужно для восстановления данных.

1 recovery blocks will be used to repair.[1 восстановительный блок будет использован для исправления]

Итак, для восстановления понадобится всего один блок. Как мы уже знаем, наибольший восстановительный файл может иметь название **F12.vol063+37.par2**; он содержит 37 блоков, начиная с 63-го. Нет нужды стартовать все блоки с первого – в случае катастрофического повреждения, требующего 30 блоков для восстановления, достаточно загрузить только этот, самый большой файл **F12.vol063+37.par2**. Если удалось по крайней мере выяснить количество восстановительных блоков, исправить повреждение скорее всего удастся.

Команда на исправление почти не отличается от проверочного варианта: достаточно вместо **par2verify** набрать **par2repair**. Проверка длилась минуту, а на восстановление понадобилось около четырёх минут. По завершении работы вы обнаружите старый файл, переименованный в **Fedora-12-x86_64-DVD.iso.1**, и восстановленный **Fedora-12-x86_64-DVD.iso**. Это немного сбивает с толку: логично предположить, что добавку «1» получит новый, восстановленный файл. В итоге контрольная сумма MD5 спасённого файла соответствует исходной, а это здорово.

Недалековидному пользователю *par2*, конечно, не поможет. Однако небольшие затраты сил и времени на постройку пирамиды восстановительных файлов многократно окупятся, когда настанет час спасать свои труды. **LXF**



» Чтобы защитить файлы, щелчком правой кнопки добавьте их в поле Files To Protect. Для проверки и восстановления просто выделите файлы *par2* и нажмите кнопку Go.

» Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.

OLAP: Анализ

Выложить белый хлеб по соседству с молоком или колбасой? Ответ на этот вопрос поможет найти многомерный анализ данных, а Вячеслав Олоничев и Андрей Сенов реализуют его на Qt и C++.



Наш эксперт

Вячеслав Олоничев

Преподает в университете UNIX IPC, разрабатывает корпоративные базы данных и приложения для промавтоматики. В Linux с 1999 года.



Наш эксперт

Андрей Сенов

Большой любитель связки C++/Qt, но не оставляет без внимания и Java. Странник Open Source.

Многомерный анализ данных или OLAP – это технология, интуитивно понятная любому, кто когда-либо занимался обобщением данных по продажам, выпуску продукции и тому подобным. Эта технология востребована практически повсеместно, начиная от маленькой мастерской или фермерского хозяйства и заканчивая министерством или центральным офисом транснациональной корпорации. Соответственно, и решений для нее существует достаточно много. Здесь можно найти и нестандартные расширения SQL типа pivot в Access, подключаемые модули для электронных таблиц или той же «1С», превеликое множество программ на Delphi с его Decision Cube или VBA, и, наконец, инструментарий для многомерного анализа на стороне сервера от Microsoft или Oracle. Наиболее удобные и продвинутое решения в данной области предназначены в первую очередь для больших корпораций и имеют соответствующую стоимость. Использование же электронных таблиц крайне трудоемко и неэффективно. И кроме того, все указанные решения являются проприетарными.

Описываемая здесь программа (вы найдете ее полные исходные тексты на LXF DVD) отличается от имеющихся в первую очередь тем, что использует только свободный код и сама является открытой. Предназначена она для многомерного анализа данных на стороне клиента, что и определяет область ее потенциального применения: небольшие и средние предприятия, для которых SQL-запрос, отбирающий данные за пять-шесть лет, будет выполнен на сервере и передан по локальной сети клиенту за приемлемое время.

Программа основана на выполнении всех операций по многомерному анализу с использованием типовых структур данных в ОЗУ, что должно обеспечить достаточно высокую эффективность при умеренных затратах памяти. Наиболее полная, удобная и эффективная реализация типовых структур данных – это параметризованные контейнеры на языке C++, такие как стандартная библиотека шаблонов STL и библиотека шаблонов Qt.

Как контейнеры STL, так и контейнеры Qt – это свободное ПО. Разница между ними настолько мала, что фрагменты исходного кода, содержащие обращение к STL, можно переделать в код, работающий с шаблонами Qt (и наоборот), буквально за пару минут.

В нашем случае используются шаблоны Qt, по следующей причине: поскольку графический интерфейс программы выполнен с использованием библиотеки Qt и доступ к базе данных производится также средствами Qt, то шаблоны Qt следует применять не только из принципа единообразия, но и ради эффективности, так как они тесно взаимодействуют с основной библиотекой. Модуль многомерного анализа в программе выделен в отдельный класс, поэтому его легко адаптировать к другой платформе или библиотеке. Кстати говоря, первая версия программы была выполнена на C++ Builder с использованием STL, потом интерфейс был переписан на Qt3 в KDevelop, но STL сохранена. А текущая реализация выполнена на Qt4 в среде Qt Creator.

Для многомерного анализа данных на стороне клиента требуется решить следующие задачи:

- 1 Выбор структуры данных для сохранения в ОЗУ результатов SQL-запроса и требования к тексту SQL-запроса.
- 2 Формат представления выбранных осей гиперкуба для формирования его проекции на плоскость.
- 3 Формирование самой проекции.
- 4 Формирование итогов.

При решении данных задач будем последовательно придерживаться принципа повсеместно использовать параметризованные контейнеры. Это стремление не столько к соблюдению единого стиля, сколько к эффективности, так как позволяет свести к минимуму все if-else'и и switch-case'и и максимально использовать оптимизированный код параметризованных контейнеров.

Выбор структуры данных

SQL-запрос, поставляющий данные для многомерного анализа, должен иметь вид:

```
select ось0, ось1, ... осьN, val
from ... Inner join ... on ...
...
where some_date between :d1 and :d2
```

где ось0, ось1, ..., осьN – результаты измерений на осях гиперкуба; val – анализируемое значение, содержащееся в ячейке гиперкуба; d1; d2 – параметры, задающие интервал времени, за который производится анализ.

Запрос возвращает реляционную таблицу, в которой атрибуты, соответствующие результатам измерений, могут иметь самые разнообразные типы, а последний атрибут – val – практически всегда является числом с плавающей точкой.

Наша задача – реализовать не только универсальный способ сохранения результатов в ОЗУ, но и обеспечить возможность по-

STL и OLAP

Прежде чем углубляться в детали, разьясим несколько терминов. Стандартная библиотека шаблонов (Standard Template Library, STL) – набор согласованных обобщенных алгоритмов, контейнеров, средств доступа к их содержимому и различных вспомогательных функций (http://ru.wikipedia.org/wiki/Стандартная_библиотека_шаблонов).

OLAP (On Line Analytical Processing – интерактивная аналитическая обработка данных) – один из способов представления и анализа данных. При этом информация представляется в виде многомерного куба с возможностью произвольного манипулирования ею. Многомерные модели рассматривают данные либо как факты с соответствующими численными параметрами, либо как текстовые измерения, которые характеризуют эти факты. (Каширин И.Ю., Семченков С.Ю. Интерактивная аналитическая обработка данных в современных OLAP системах // Бизнес-информатика – 2009. – № 2. С.12–19. См. также <http://ru.wikipedia.org/wiki/OLAP>).

ДАННЫХ И СПО

строения как простых, так и составных ключей по результатам измерений на осях гиперкуба, которые будут необходимы как для упорядочивания, так и для поиска информации. Поэтому все результаты измерений следует привести к одному типу в самом начале, еще до размещения в ОЗУ. В качестве такого типа был выбран строковый. Это объясняется тем, что большинство атрибутов, соответствующих результатам измерений на осях гиперкуба, изначально имеют строковый тип – это наименования товаров, предприятий, фамилии работников и тому подобное. Все другие типы всегда можно преобразовать к строковому. Что касается дат, то если их преобразовать в строковый вид при помощи SQL-функции **cast** в так называемом армейском формате – «гггг.мм.дд» – то упорядочивание строк в алфавитном порядке будет соответствовать упорядочиванию дат в порядке возрастания.

Таким образом получаем:

```
typedef QVector<QString> s_keys;
struct l_cube{
    s_keys dims;
    double r;
};
typedef QVector<l_cube> data_cube;
```

где `s_keys` – вектор, в который помещаются результаты измерений; `l_cube` – структура для сохранения одной строки, возвращаемой запросом; `data_cube` – вектор, для сохранения в ОЗУ таблицы, возвращаемой запросом.

Закачка информации в память, если смотреть изнутри, осуществляется следующим образом:

```
class hyper_cube{
protected:
    data_cube dc;
    l_cube line;
...
public:
    void add(s_keys sk, double res){
        line.r = res;
        for(int i=0; i<N; i++){
            line.dims[i]=sk[i];
            dc.push_back(line);
        }
...
}
```

А снаружи это выглядит так:

```
hyper_cube HC;
s_keys VK;
QStringQuery* psql;
QDate d1,d2;
psql->bindValue(":d1",QVariant(d1));
psql->bindValue(":d2",QVariant(d2));
if(psql->exec()){
    while ( psql->next() ) {
        for(int i=0; i<Nax; i++){
            VK[i] = psql->value(i).toString() + '\n';
        }
        HC.add(VK, psql->value(Nax).toDouble());
    }
}
```

Здесь `Nax` – количество осей гиперкуба. Если использовать однонаправленный запрос, то приведенный код будет достаточно эффективным.

База данных

В качестве примера используем фрагмент базы данных предприятия, занимающегося мелкооптовой продажей столовой посуды. База данных реализована на SQLite (каталог `primer` в исходных текстах программы), а ее ER-диаграмма представлена на рис. 1.

В указанной базе данных имеются следующие таблицы: `kat` – категории товара, `tov` – товары, `prod` – продавцы, `grup` – группы покупателей, `pok` – покупатели; `cen` – цены на товары, `sales` – продажи.

Следующий запрос возвратит гиперкуб с данными о продажах в рублях, содержащий 8 осей:

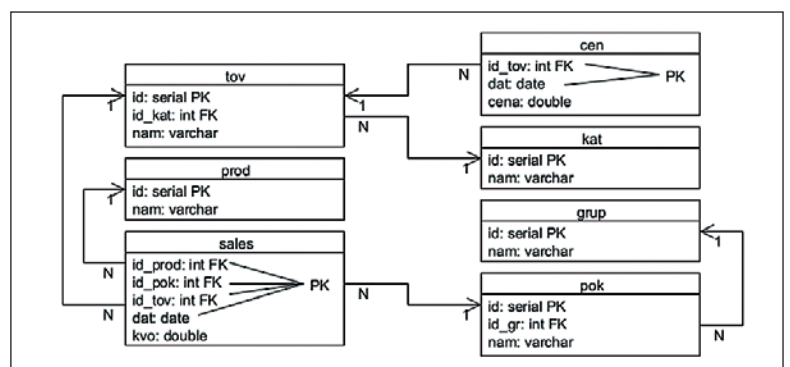
```
select k.nam, t.nam, pr.nam, g.nam, pk.nam,
substr(cast(s.dat as char(32)),1,4) as ye,
substr(cast(s.dat as char(32)),1,7) as mo,
substr(cast(s.dat as char(32)),1,10) as da,
s.kvo*c.cena as sum
from sales s inner join cen c on c.id_tov=s.id_tov
inner join tov t on t.id=s.id_tov
inner join prod pr on pr.id=s.id_prod
inner join pok pk on pk.id=s.id_pok
inner join grup g on g.id=pk.id_gr
inner join kat k on k.id=t.id_kat
where c.dat=(select max(dat) from cen where
cen.id_tov=s.id_tov and dat<=s.dat)
and s.dat between :d1 and :d2
```

Осями гиперкуба в данном случае являются: 0 – категория, 1 – товар, 2 – продавец, 3 – группа, 4 – покупатель, 5 – год, 6 – месяц, 7 – день. Фрагмент гиперкуба или таблицы, полученной в результате выполнения данного запроса, можно видеть на рис. 2.

Формат представления

Следующая проблема – выбор пользователем осей гиперкуба для формирования проекции и передача этой информации коду, который и будет ее формировать. Для пользователя естественным представлением осей гиперкуба являются два списка, один для оси X проекции и второй – для оси Y, допускающих множественный выбор и перемещение элементов списка относительно друг

► Рис. 1. ER-диаграмма базы данных «Оптовая торговля столовой посудой».



name	character	year	text	no	da	su	sum	
серебро	лодка	Коска	Восток	ООО Прогресс	2007	2007-07	2007-07-13	4920
серебро	лодка	Пава	Иг	ООО Дельта	2007	2007-07	2007-07-10	0604
серебро	лодка	Даца	Запад	ООО Полюна	2007	2007-07	2007-07-28	912
серебро	лодка	Поша	Север	ООО Орион	2007	2007-07	2007-07-21	5688
серебро	лодка	Даца	Север	ООО Сатурн	2007	2007-07	2007-07-24	1344
серебро	лодка	Коска	Запад	ООО Сосна	2007	2007-07	2007-07-25	1728
серебро	лодка	Даца	Восток	ООО Ракета	2007	2007-00	2007-00-01	5140
серебро	лодка	Даца	Север	ООО Сатурн	2007	2007-00	2007-00-04	5472
варфор	блюдце	Даца	Восток	ООО Комета	2007	2007-07	2007-07-14	4158
варфор	блюдце	Даца	Запад	ООО Сосна	2007	2007-07	2007-07-17	1585
варфор	блюдце	Даца	Восток	ООО Прогресс	2007	2007-07	2007-07-18	1425
варфор	блюдце	Пава	Иг	ООО Ганга	2007	2007-07	2007-07-20	4015

Рис. 2. Фрагмент гиперкуба с данными о продажах в рублях, содержащего восемь осей.

друга. Эти два требования являются обязательными; программа, не отвечающая им, не представляет большого интереса для практики.

Для приложения использование наименований осей не совсем удобно – предпочтительней использовать номера: ведь данные измерений хранятся в векторах, и индекс в векторе однозначно соответствует номеру оси. Поскольку элементы списка могут изменять взаимное расположение, то их текущие номера уже не будут соответствовать номерам осей гиперкуба. Для решения данной задачи воспользуемся словарем `QMap<QString, int>`, который позволит по выбранной пользователем строке с названием оси определить ее номер. Допустим, что, как представлено на рис. 3, пользователь решил проанализировать, как распределяются продажи по годам и группам покупателей, с одной стороны, и по категориям товаров и продавцам, с другой. Для этого ему пришлось переместить ось гиперкуба «Год» вверх по списку для проекции на ось X – ведь его интересует распределение продаж по группам покупателей в пределах каждого года; иначе у него получилось бы распределение продаж по годам в пределах каждой группы.

Определим следующие переменные: `myAxes` – массив `QVector<QString>`, содержащий наименования осей гиперкуба; `lstX` и `lstY` – списки `QListWidget`, предназначенные для отображения и выбора осей гиперкуба для осей X и Y проекции; `m_axe` – словарь `QMap<QString,int>`, позволяющий получить порядковые номера по наименованиям осей гиперкуба; `xJ`, `yI` – массивы `QVector<int>`, содержащие номера осей гиперкуба, отобранные для отображения на оси X и Y проекции. В результате, в примере на рис. 3, массив `xJ` будет содержать числа 5 и 3, а массив `yI` – числа 0 и 2.

Формирование проекции

При формировании отчета будем использовать следующие правила: в ячейки проекции помещаются только суммы, а итоги могут содержать или суммы, или средние, или минимумы/максимумы значений, содержащихся в клетках проекции, в зависимости от выбора пользователя. Таким образом, при выборе максимума мы будем получать максимальную сумму продаж категории товаров продавцом каждой группе потребителей за год, а не максимальную единичную продажу.

Как проекцию, так и итоги будем помещать в двумерный массив чисел с плавающей точкой. Значения измерения по осям «пририсовываются» к отчету на последнем этапе, когда он отображается в `QTableWidget` в окне диалога. Размер данного двумерного массива известен на момент формирования, и поэтому мож-

но использовать обычный динамический массив. Однако, просто из принципа единообразия, будем использовать тип данных `QVector<QVector<double>>`.

На рис. 4 представлен фрагмент проекции и итогов; заголовки колонок таблицы содержат результаты измерений по осям гиперкуба – «Год» и «Группа предприятий», которые отражены на ось X проекции. В клетках проекции представлены суммы продаж, распределенные по годам и группам предприятий, с одной стороны, и категориям товаров и продавцам, с другой стороны. Итоги внизу представляют суммы по годам и группам предприятий. Строчка ниже содержит суммы продаж по годам, а самая правая колонка – суммы продаж по категориям товаров.

Задача заключается в том, чтобы выбрать из таблицы с результатами запроса уникальные сочетания измеренных значений на выбранных осях гиперкуба и упорядочить их в алфавитном порядке. А затем следует отобрать значения анализируемой величины, в данном случае – суммы продаж, из соответствующих строчек таблицы и просуммировать их в нужных клетках проекции. Для решения задачи будем использовать словарь `QMap<QString, int>`.

В данном фрагменте кода сканируется таблица `dc`, содержащая результаты запроса к базе данных, как показано на рис. 2. Векторы `xJ` и `yI`, как уже было сказано, содержат номера выбранных осей, {5, 3} и {0, 2} соответственно. В ходе выполнения цикла строка `xs` будет последовательно принимать значения «2007\nВосток\n», «2007\nЮг\n», «2007\nЗапад\n», «2007\nСевер\n» и т.д., а строка `ys` – «серебро\nКосюша\n», «серебро\nМаша\n», «серебро\nДаца\n», «серебро\nМаша\n» и т.д. И все эти строки, являющиеся полными составными ключами для осей X и Y проекции гиперкуба, будут помещаться как ключи в словари `pX` и `pY`. Значения, помещаемые в словари, на данном этапе не важны, и поэтому для них используется -1. Словарь может содержать только неповторяющиеся ключи, поэтому по завершении цикла словари `pX` и `pY` будут содержать все уникальные значения полных составных ключей, содержащихся в таблице с данными.

```

QString xs, ys;
Qmap<QString,int> pX,pY;
int Hx=xJ.size();
int Hy=yI.size();
for(int i=0; i<dc.size(); i++){
    xs=""; ys="";
    for(int x=0; x<Hx; x++) xs=xs+dc[i].dims[xJ[x]];
    for(int y=0; y<Hy; y++) ys=ys+dc[i].dims[yI[y]];
    (*pY)[ys]=-1; (*pX)[xs]=-1;
}
    
```

Следующий фрагмент кода перебирает все ключи, содержащиеся в словарях `pX` и `pY`, при помощи итератора, и присваивает им монотонно возрастающие целочисленные значения. В результате наши словари содержат в качестве ключей наименования строк и колонок проекции, а в качестве значений – номера соответствующих строк и колонок. Это как раз то, что и требуется для заполнения ячеек проекции.

```

Qmap<QString,int>::iterator it;
int m,n;
m=0;
for(it=(*pY).begin(); it!=(*pY).end(); it++){
    {it.value()=m++;}
n=0;
for(it=(*pX).begin(); it!=(*pX).end(); it++){
    {it.value()=n++;}
    
```

В нижеследующем цикле значения полных составных ключей опять восстанавливаются из значений измерений, а по ним определяются координаты ячейки проекции, к содержимому которой добавляется значение анализируемой величины из очередной строчки сканируемой таблицы с результатами запроса.

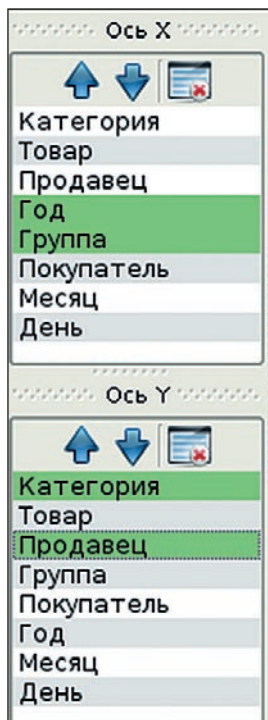


Рис. 3. Выбор осей гиперкуба для формирования проекции.

	2007 Восток	2007 Запад	2007 Север	2007 Юг	2009 Юг	Сумма	
серебро Даша	1527696.00	1294119.00	1018791.00	793728.00	1126692.00	13769532.00	
серебро Ксюша	1421793.00	862200.00	891594.00	1096515.00	1033164.00	14989626.00	
серебро Маша	561195.00	1228338.00	841554.00	934425.00	925524.00	12356712.00	
серебро Паша	377757.00	1195623.00	1106604.00	847971.00	1297521.00	12732624.00	
серебро Саша	1017108.00	1472616.00	955728.00	377208.00	1355040.00	13408551.00	67257045.00
хрусталь Паша	3238830.00	2901960.00	4131441.00	3484872.00	2358378.00	39758418.00	
хрусталь Саша	2835360.00	2052648.00	3350673.00	3924603.00	3168261.00	37068939.00	188951508.00
Сумма	22007709.00	22890006.00	24154362.00	25072092.00	27594108.00	310583610.00	
				94124169.00	115448841.00		

Рис. 4. Фрагмент отчета, генерируемого программой.

```
int I,J;
for(int i=0; i<dc.size(); i++){
  xs=""; ys="";
  for(int x=0; x<Hx; x++) xs=xs+dc[i].dims[xJ[x]];
  for(int y=0; y<Hy; y++) ys=ys+dc[i].dims[yI[y]];
  I = (*pY)[ys];
  J = (*pX)[xs];
  dp[I][J]+=dc[i].r;
}
```

Формирование итогов

Осталась последняя проблема – формирование итогов по полным и частичным ключам. Как уже было сказано выше, надо получить суммы продаж, распределенные по группам потребителей в пределах года, а также суммы продаж по годам. Так вот, «Год\nГруппа_компаний\n» – это полный ключ, имеющий уникальное значение для каждой колонки проекции, а «Год\n» – это частичный ключ. Если построить словари с частичными ключами, то можно определить как координаты ячеек в итогах, «пристроенных» к проекции справа и снизу, так и расстояние между этими координатами – то есть количество ячеек в итогах, определенных по полным ключам, которые необходимо использовать для формирования итогов по частичным ключам. Поскольку количество осей гиперкуба, накладываемых на ось проекции, может быть произвольным, то для формирования частичных ключей будем использовать вектор словарей: `QVector<QMap<QString,int>>`.

Формирование итогов будем осуществлять при помощи циклов, перебирающих ячейки массива с проекцией гиперкуба. А проблему вида итогов, а именно: сумма, среднее, максимум или минимум – будем решать при помощи виртуальных функций вспомогательных классов.

```
class Ttl{
protected:
  double r;
public:
  void setR(double R);
  virtual void updR(double E)=0;
  virtual double getR(int N)=0;
};
class Sum: public Ttl{
public:
  virtual void updR(double E);
  virtual double getR(int N);
};
class Avg: public Sum{
public:
```

```
virtual double getR(int N);
};
class Max: public Sum{
public:
  virtual void updR(double E);
};
class Min: public Sum{
public:
  virtual void updR(double E);
};
void Ttl::setR(double R){r=R;}
void Sum::updR(double E){r+=E;}
void Max::updR(double E){if(E>r)r=E;}
void Min::updR(double E){if(E<r)r=E;}
double Sum::getR(int N){return r;}
double Avg::getR(int N){return r/N;}
```

Имена у классов говорящие, а предназначение функций очевидно и не требует разъяснений. Единственное, что вызывает вопрос, так это виртуальные функции. Стоило ли бороться за эффективность кода, используя параметризованные контейнеры, чтобы потом все, сэкономленное буквально по миллисекундам, так бездарно спустить в последний момент? Поясним: параметризованные контейнеры используются внутри циклов, сканирующей таблицу с исходными данными или гиперкуб. Данная таблица в реальных задачах может иметь несколько десятков, а то и сотен тысяч строк. А для массива, в котором формируется проекция и итоги, счет идет на десятки. Теоретически, конечно, можно сформировать отчет и на несколько тысяч строчек и колонок, только кто же его смотреть будет?

Финальные штрихи

Вывод проекции и итогов на форму и добавление к ним заголовков строк и столбцов – самостоятельная задача, и в данной статье не рассматривается, наряду с остальными вопросами проектирования интерфейса с пользователем, а также формата файла, в который помещается отчет, сгенерированный программой. Но в программе, которую (напомним) вы найдете на **LXF DVD**, все это сделано. Поэтому она готова к использованию «из коробки».

Предлагаемая методика многомерного анализа данных, конечно, не ограничивается только языком C++ и библиотекой Qt. Вполне возможна реализация на wxWidgets с использованием STL, а параметризованные шаблоны имеются и в языках Java и C#.

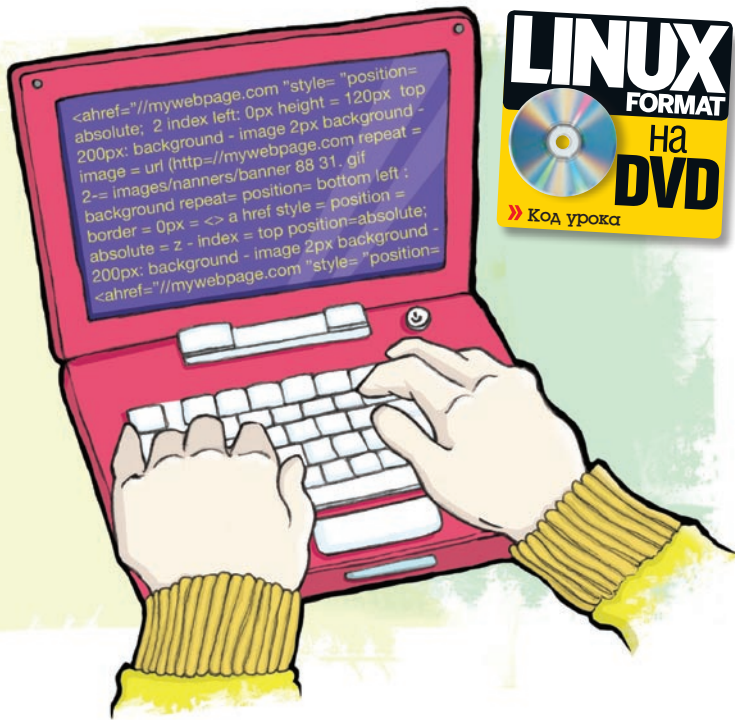
В последний момент, просматривая демонстрационную базу данных, мы заметили, что девушки-продавщицы проработали у нас по три года без отпусков. Да... Лучше бы мы взломали базу данных сиротского приюта. **LXF**

» Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.



Python: Будем

Познакомьтесь с прекрасным инструментом для создания и выпуска приложений от команды Ubuntu. Ник Вейч настраивается на любимый подкаст...



Что нам понадобится

Наш урок основан на *Quickly*, инструменте разработки Ubuntu; поэтому будет намного проще, если вы возьмете Ubuntu (например, с диска LXF133). Тогда просто установите пакеты *quickly* и *python-feedparser* из репозитория, и все будет готово.

Если вы распаковали код с LXF DVD в удобное место, можете использовать для урока его – настроенное *Quickly* уже готово к работе, и можно использовать команды *Quickly*.

стными лентами и тому подобным, но для начала припасем место для хранения загружаемых файлов. Мы обязаны дать пользователям возможность выбрать место хранения (или создать новое, если надо); значит, необходим диалог, в котором они смогут это сделать.

Весьма удобно, что шаблон нашего приложения содержит редактор параметров. В нем еще ничего нет, но это уже хорошее начало. В таких ситуациях обычно лучше начать с интерфейса пользователя, чем с самого кода – хотя бы потому, что это поможет узнать имена объектов и их сигналы до написания кода, объединяющего их. Итак, запустите *Glade* командой *quickly design* в каталоге приложения.

Откроется *Glade* с главным UI-файлом проекта, но нам-то нужен другой, так что откройте правильный, с именем *PreferencesMyAppDialog.ui*. После его открытия вы увидите посередине приятную пустоту. Первым делом разместим в ней элемент *vbox*, вертикальный контейнер – он размещает свои дочерние элементы по вертикали. Щелкните в левой панели на иконке *vbox*, а затем в серой области, чтобы разместить его. Всплывет диалог, запрашивающий, сколько элементов вы хотите разместить – оставьте их 3. Пока мы употребим только два из них, но это не беда – один будет про запас, поскольку, вероятнее всего, он понадобится нам позднее. Но не запасайте слишком много: добавить элементы в вертикальный контейнер просто, а вот удалить – большая морока.

В верхней части разместим метку, чтобы пользователь понимал, что происходит. Выберите элемент *Label* в панели слева и щелкните в верхней части *vbox*. Сейчас в ней написано просто «Label», но текст можно изменить при помощи инспектора свойств справа внизу.

Поместив более осмысленный текст в свойстве Метка, добавьте какой-нибудь способ выбора каталога. Таковых можно придумать много, но *GTK* содержит специальную кнопку именно для этой цели, и было бы глупо ею не воспользоваться. Выберите объект *FileDialogButton* в панели слева и поместите его в срединный слот.

Нам нужно изменить некоторые свойства. На вкладке «Основные» прокручивайте список вниз, пока не наткнетесь



Наш эксперт

Ник Вейч редко делает что-нибудь Быстро, так что учебник этого месяца – нечто вроде проверки себя.

На этом эпическом и зрелищном уроке наша задача, ни много ни мало, написать полноценное настольное приложение, создать для него пакет Debian и выпустить его в большой мир. Думаете, невозможно создать полноценное работающее настольное приложение, которое и вправду что-то делает, в рамках четырехстраничного урока? В чем-то вы правы; вот мы и расширили урок этого номера в два раза, до восьми страниц. И на сей раз у нас есть сильный союзник – *Quickly*.

Quickly [англ. «быстро»] – это набор средств разработки, с прилагаемыми шаблонами, для создания подобия RAD-среды (Rapid Application Development – быстрая разработка приложений). Оно довольно скоростное: можно создать и запустить приложение за несколько секунд – просмотрите пошаговое руководство на соседней странице и убедитесь наглядно. Однако, хоть мы и можем создать работающее приложение, в нем не будет ни грамма функционала, необходимого для мегазагружчика подкастов *TuxRadar* (а его-то мы и собираемся делать). И все же пробежитесь по тексту, чтобы понять, как все работает, затем вернитесь сюда – и мы разберемся с подкастами.

Создаем приложение

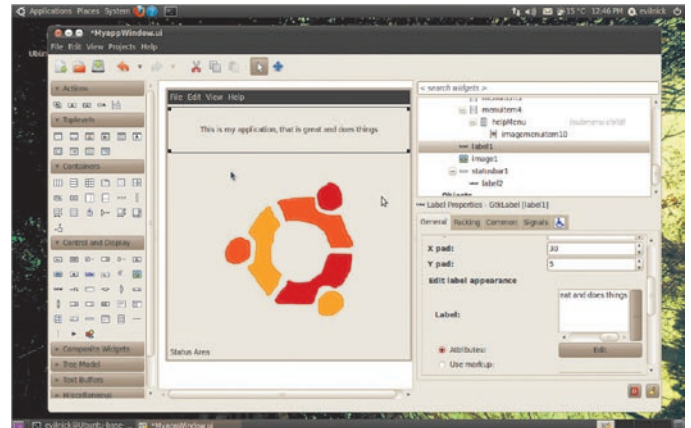
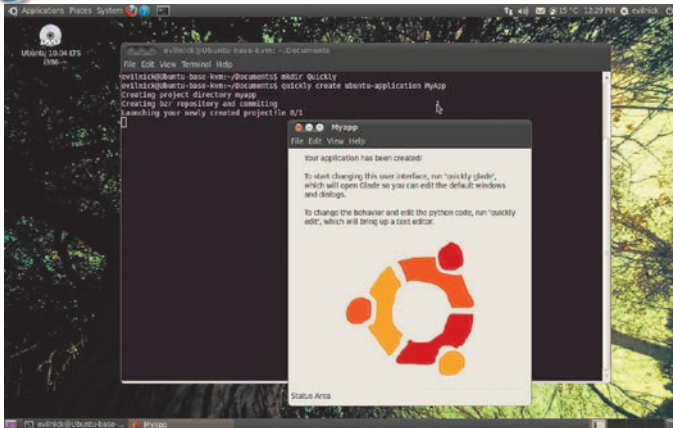
Итак, мы ознакомились с базовыми механизмами работы *Quickly*; пора сделать что-то полезное. Как уже сказано, мы намерены создать загрузчик подкастов. Понятно, придется повозиться с ново-

» Месяц назад Мы приняли снежинки Коха в актеры *Clutter* с помощью *Cogl*.

ГРУЗИТЬ ПОДКАСТЫ



Шаг за шагом: Создаем простое приложение Quickly

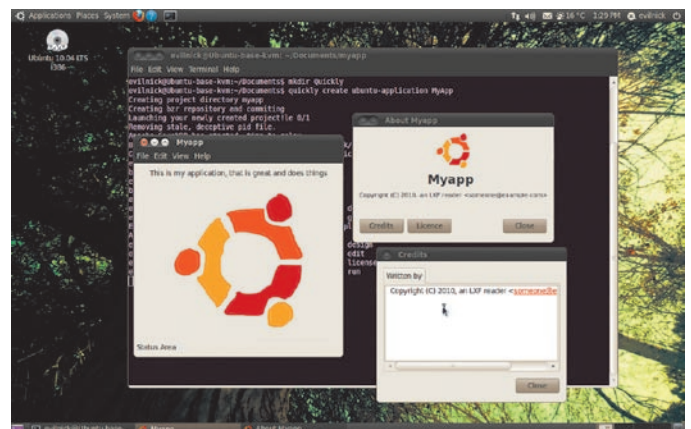
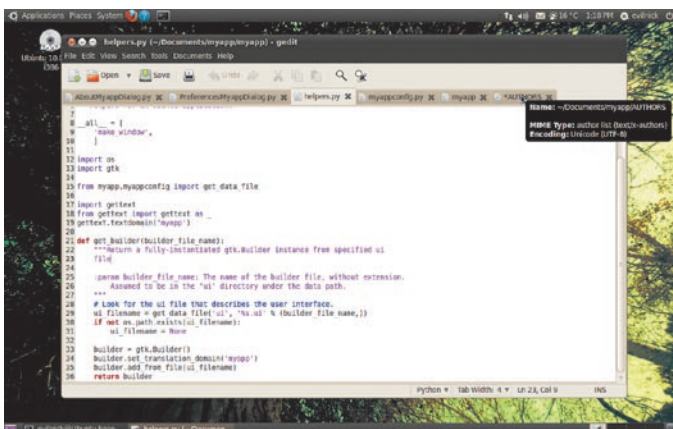


1 Создаем в Quickly

Чтобы начать создавать свое приложение, откройте терминал и создайте новый каталог для всех ваших Quickly-проектов командой `mkdir Quickly`. Войдите в этот каталог (`cd Quickly`) – и вы готовы к созданию и сборке вашего первого приложения. Введите `quickly create ubuntuapplication MyApp`, и все закрутится. *Quickly* создаст каталог и сгенерирует в нем все необходимые файлы, а именно – файлы интерфейса *GTK*, файлы Python, базы данных и системы управления версиями. По завершении приложение запустится.

2 Проектируем в Quickly

Войдите в каталог проекта `myapp` и скомандуйте `quickly design`. Запустится *Glade*, редактор графических интерфейсов для *GTK*, и в нем автоматически откроется главное окно вашего приложения. Все файлы интерфейса пользователя хранятся в одном и том же месте, так что любой диалог вашего приложения можно легко изменить. Немного порезвьтесь в *Glade*, если ранее вы с ним не сталкивались. Свойства каждого объекта легко изменить при помощи инструментов, расположенных в правом нижнем углу. Здесь мы меняли текст в нашей метке. Не забудьте сохранить проект перед выходом!

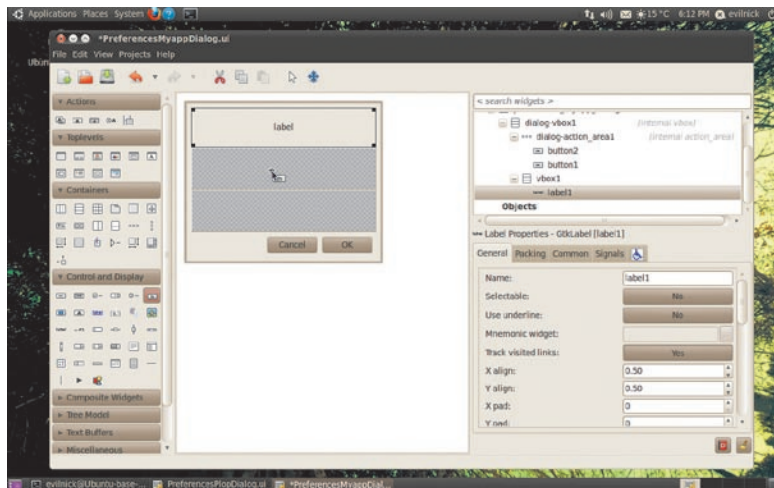


3 Редактируем в Quickly

Quickly использует редактор *Gedit* из *Gnome*, прекрасно подходящий для неинтенсивной работы. Кроме того, он отслеживает файлы вашего проекта, поэтому для их изменения, находясь в терминале в главном каталоге приложения, просто введите `quickly edit`. *Gedit* появится и откроет все Python-файлы, связанные с проектом, так что вы можете начать создание новых функций. Из других файлов вы можете пожелать изменить файл `AUTHORS` в главном каталоге проекта. Просто введите краткое сообщение об авторе, чтобы все знали, что это ваше. Сохраните файл и выйдите.

4 Запускаем в Quickly

Перед тем, как вы запустите свое слегка измененное приложение, сделайте еще кое-что. В командной строке введите `quickly license`. Это создаст сообщение об авторском праве на основании той информации, что вы ввели в файле `AUTHORS`, и уведомление *GPL v3* (по умолчанию, хотя это можно изменить) в заголовке каждого файла с исходными кодами; ну, и еще там всякое. Для запуска вашего приложения введите `quickly run`. Когда оно откроется, выберите в меню `Help > O` программе. Вы увидите, что диалог изменился и теперь содержит ваше сообщение об авторском праве и информацию о лицензии!



Элементы интерфейса можно выбирать либо на самой форме, либо в древовидном списке справа. Содержимое инспектора свойств изменится в соответствии с выбранным элементом, и можно будет редактировать его поведение.

на пункт «Действие». Измените его на «Выбор каталога». Стандартное действие – это выбор файла, но это не то, что нам нужно. После выбора этого действия, щелчок на кнопке приведет к заполнению списка соответствующим содержимым. Мы можем опробовать ее уже в *Glade* – просто щелкните на иконке у правого края кнопки.

Однако прежде чем сохранить наш новый диалог, сделайте еще одну важную вещь. Когда кто-нибудь выбирает каталог, мы должны как-то узнать, что же он выбрал. Для этого мы подаем сигнал от объекта, сообщающий, что некое значение изменилось. С точки зрения кода, нужно написать метод-обработчик, работающий с данными, но обработчик необходимо указать здесь при дизайне UI-файла. Щелкните на вкладке «Сигналы». Вы увидите, что сигналов, к которым можно подключиться, полно. Нас интересует сигнал из раздела 'GTKFileChooser' – откройте его. В первом столбце дважды щелкните справа от названия 'current-folder-changed' и введите имя обработчика 'folder_changed'. Теперь, стоит пользователю выбрать другой каталог, мы узнаем об этом и изменим соответствующий параметр приложения.

Не забудьте сохранить файл перед выходом из *Glade*. Теперь пришло время ввести **quickly save 'prefs dialog'**, что передаст ваши новые файлы системе управления версиями (см. совет *Quickly Сохраняйте почаще*).

Храните здесь

Прежде чем погрузиться в код, чтобы сделать этот шедевр интерфейса рабочим, прикинем, где мы будем все размещать: ведь главный смысл настроек в том, что они сохраняются на будущее. Так что требуется место для хранения. Данные будут типа настроек приложения, лент, на которые мы подпишемся, и, возможно, многого другого, о чем мы еще не задумывались. Лучшее всего выбрать нечто легкодоступное. Имеется множество популярных и хорошо документированных модулей Python для сохранения

Что в имени твоём?

Не исключено, что вы считаете UI-файлы *Glade* просто XML-кодом, однако вы не сможете просто перенести их из одного проекта в другой без значительного рефакторинга. Они напичканы ссылками на имя проекта, так что вы не сможете даже просто импортировать их в ваш проект – лучше позволить *Quickly* создать для вас базовые UI-файлы.

настроек, вроде **ConfigObj**, **ConfigParser** и т.д., но мы используем нечто иное: базу данных под названием *CouchDB*.

Возможно, вы сочтете, что база данных для хранения пары параметров – это перебор, но в ее пользу есть веский довод: она уже является частью шаблона проекта, который *Quickly* использует для генерации нашего приложения, а коли она есть, надо ею пользоваться.

На самом деле в проект включена не только база, но и код для доступа к ней; остается добавить там и сям несколько строк. Если вы желаете сперва ознкомиться с *CouchDB*, просмотрите врезку *Что такое CouchDB?*

Запустите *Gedit* командой **quickly edit**, и все файлы с кодом откроются автоматически. Найдите **PreferencesMyAppDialog.py** с кодом обработки настроек. Неважно, понимаете ли вы весь этот код или нет, но чтение его вам поможет. Здесь происходит несколько вещей: несколько функций инициализации, генерирующих окно диалога, и некий код для работы с базой данных. К счастью, в нем имеются комментарии, и найти его будет не сложно. При выполнении данный код проверяет, не заданы ли уже некоторые настройки; если нет, он открывает базу данных (и запускает *CouchDB*, если она еще не запущена) и пытается загрузить их из нее. Если там их нет, необходимо ввести в код установку значений по умолчанию, что мы далее и сделаем. Здесь есть небольшой код для кнопок 'OK' и 'Cancel', которые вскоре появятся.

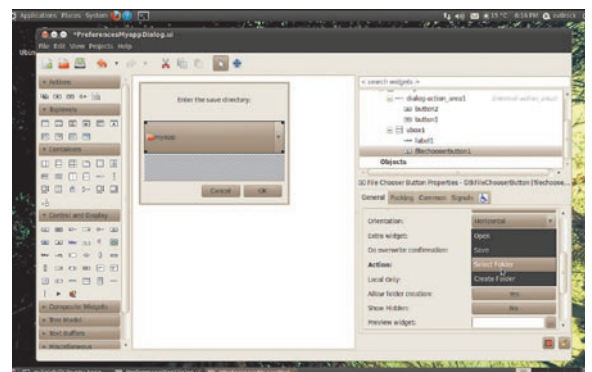
Сохраняем вводимые значения

Итак, мы решили сохранить в настройках каталог, куда будут загружаться файлы. Согласно коду, настройки хранятся в объекте-словаре с именем **self.preferences** (часть **'self'** означает, что это часть данных для объекта-диалога, создаваемого при вызове этого кода). Здесь есть даже подходящее место для размещения умолчаний – в методе с именем **_load_preferences**, а также имеется комментарий TODO, сообщающий, что здесь-то и следует разместить значения по умолчанию. Здесь и введите

```
self.preferences["savedir"]="~/"
```

(Убедитесь, что в коде должно количество отступов, иначе получите ошибку). Это кусочек магии сокращений в Linux. Символ «тильда» (~) в записи пути к файлу распознается как путь к домашней директории пользователя, так что эта строка автоматически устанавливает его в качестве каталога сохранений, если пользователь не изменит это.

Теперь добавим обработчик сигнала, который мы ввели при разработке интерфейса пользователя. Разместить его можно в любом месте класса, но, наверно, лучшим вариантом будет непо-



Сделайте действием кнопки файлового диалога «Выбор каталога», иначе пользователи смогут выбирать во всплывающем списке только файлы.

» Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.

средственно перед строкой, начинающейся с `def ok(self, widget, ...` – обработчиком нажатия клавиши ОК:

```
def folder_changed(self, widget, data=None):
    self._preferences["savedir"]=widget.get_current_folder()
```

Все очень просто. В *PyGTK*, виджет, подающий сигнал, включает самого себя в обратный вызов обработчика. Как мы знаем, это сигнал от *FileChooserDialog*; мы также знаем, что у него есть метод добывания текущего выбранного значения (в виде строки), и можно просто сохранить их в нашем словаре настроек.

Тут, однако, встает интересный вопрос: с каким значением появляется виджет? Ответ таков: если мы ранее ничего не определили, то запускается он с пустым значением. Это может обескуражить пользователя; нужно заранее включить некоторый код, чтобы корректно задать значение. Для этого потребуется раздобыть ссылку на сам виджет.

Код в нашем файле использует самостоятельный сборщик приложения для создания реального диалога из файла `.ui`, который мы сохранили, так что на графический интерфейс ссылаются локально через объект `self.builder`. Имя виджета мы помним (`filechooserbutton1`), и можем просто разыскать этот объект и сохранить в нем переменную с настройками при помощи подходящего метода:

```
o=self.builder.get_object('filechooserbutton1')
o.set_current_folder(self._preferences["savedir"])
```

Лучше всего разместить это где-то сразу после создания диалога и загрузки настроек. К счастью, вы вновь обнаружите полезную заметку где-то в строке 60: **#TODO: code for other initialization actions should be added here** [#Сделать: код других действий инициализации следует разместить здесь], так что добавьте его сразу после нее (точный номер строки зависит от других функций, таких как генерация лицензии и прочее – просто призовите свой интеллект, чтобы найти ее!).

Итак, мы создали место для хранения настроек, привязали сигнал к методу получения нового значения при его изменении пользователем и не забыли взять значение и поместить его в виджет при открытии диалога. Код для кнопки ОК позаботится о сохранении любых изменений. Надо еще что-нибудь?

Сброс CouchDB

Когда наше приложение ищет сохраненные настройки при старте, оно послушно загружает то, что находит. Но если вы начнете добавлять в разные вещи код, их может не оказаться в базе данных, и ничего не произойдет. В этом случае проще всего удалить базу данных *CouchDB* из браузера. Перейдите в `file:///home/<имя_пользователя>/local/share/desktop-couch/couchdb.html` и просто выделите и удалите базу данных вашего приложения. При следующем запуске она будет создана из стандартных значений.

Что произойдет, если пользователь изменит каталог, а затем передумает и нажмет кнопку Cancel [Отмена]? Согласно текущему коду, настройки изменяются (но не сохраняются) в момент, когда пользователь выбирает другой каталог. Если он нажмет Cancel, то каталог хотя и не сохранится, но останется в памяти как выбранный параметр настройки. Следует учесть это и дать знать основному приложению об изменении параметров, просто перезагрузив их из базы данных при закрытии диалога. Для этого нужно изменить главный файл *MyApp.py* (ну или как вы там его назвали), а также обеспечить, чтобы настройки загружались при старте приложения. Код, выполняющий это, практически тот же, и не будь он всего из двух строк, его следовало бы оформить в виде функции:

```
dlg = PreferencesPlopDialog.PreferencesPlopDialog()
self.preferences = dlg.get_preferences()
```

Здесь просто создается (но не показывается) диалог, а затем из него берутся настройки. Место для этого кода в основном файле – приблизительно около строки 90 (после комментария **# Code for other initialization actions should be added here** [остальной код инициализации разместить здесь]) и строки 103 (после комментария **# Make any updates based on changed preferences here** [Любые изменения на основе параметров настройки выполнять здесь]).

Этого должно хватить. Теперь запустите приложение и посмотрите, как оно работает. Протестируйте выбор каталога и закройте и откройте приложение вновь, прежде чем мы двинемся дальше.

Что за штука — CouchDB?

» Итак, я понимаю, что *CouchDB* — еще одна база данных, как SQL и всякое такое?

» Не совсем. *CouchDB* — скорее движок хранилища документов. В отличие от базы данных, он просто хранит сведения, которые вы ему передаете, почти их не структурируя, легким для доступа способом. Жесткой схемы данных нет (все организовано в виде пар **ключ:значение**, по типу объекта-словаря в Python), что весьма удобно для нас.

» Он может хранить только текст? А как насчет других данных, которые я захочу туда поместить?

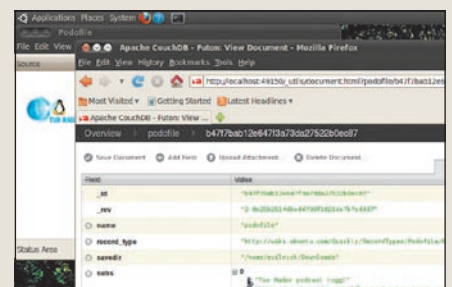
» Значение ключа может быть практически любым типом данных Python — числом, строкой и набором различных объектов, списком и даже другим словарем. Вы можете добавлять другие объекты — например, файлы с изображениями — в качестве приложений к базе данных, но более эффективно хранить их другим способом.

» Значит, тот, кто возьмет мое приложение для своей работы, должен запустить еще один сервер базы данных?

» Нет, об этом в Python позаботится библиотека *DesktopCouch*. Она запускает базу данных по любому требованию. *CouchDB* разработан нетребовательным к ресурсам, так что скорее всего вы даже не почувствуете, что он запустился. Если, конечно, вы не следите за ним...

» То есть как это — «следить за ним»?

» Ну, одно из самых замечательных свойств этой базы данных — то, что она создана с прицелом на web. Она была разработана для работы с несложными API, реализованными через HTTP, чтобы сделать ее полезной для web-приложений. Базируясь на принципах ACID (для фанатов аббревиатур: Atomicity, Consistency, Isolation и Durability — Атомарность, Согласованность, Изолированность, Долговечность), она создавалась с целью быть постоянно в рабочем состоянии, чтобы сопротивляться сбоям или отключению. Она может выполнять всякие полезные действия, вроде P2P-репликаций. Для наших целей, следует лишь направить браузер по адресу `file:///home/<имя_пользователя>/local/share/desktop-couch/couchdb.html` — там



» *CouchDB* — движок базы данных для приложений *Quickly* по умолчанию, очень удобная во многих отношениях.

вы обнаружите все базы данных, используемые *desktop-couch*.

» Выходит, это нечто вроде *MySQL-Admin*?

» Похоже, только работать с этим проще. Использование web-интерфейса для изменения данных в любой базе весьма просто и, конечно же, отслеживать транзакции базы данных также легко.



➤ Ради этого стоило попотеть — теперь можно слушать британскую команду LXF, когда только захочется.

OK, до сих пор мы вас щадили, поскольку вы, скорее всего, новички в *Quickly*, но теперь вы знаете дорогу, и нам следует ускориться, если мы хотим завершить наш загрузчик подкастов — уж извините, каждое нажатие кнопки разжевывать не будем. Если вы в чем-то не уверены, всегда можно обратиться к файлам проекта на DVD!

Наша следующая задача — создать основной интерфейс, так что наберите **quickly design** и загрузите соответствующий файл. Удалите имеющиеся там объекты, но вертикальный контейнер можно оставить.

Нам нужен способ показать список источников и список файлов выбранного источника. Это можно сделать по-разному, но в нашем приложении мы воспользуемся просмотром в горизонтальной панели, чтобы два списка разделялись вертикальной чертой. Действуйте: разместите ее в пустом контейнере.

Внутри каждой панели следует поместить контейнер Прокручиваемое окно (возьмем его в том же разделе палитры слева), это поможет просматривать списки большой длины.

Теперь в правой панели разместите объект Дерево. Это похоже на список, и мы будем применять его для отображения графики, идентифицирующей подкасты, на которые мы подписаны. При размещении объекта Дерево *Glade* потребует указать его модель. Просто создайте новую. Модель — это обычный список типов данных и меток, использующихся для манипуляций с данными или отображения их в интерфейсе пользователя. Мы начнем с четырех значений — `guint` (целое без знака) с именем `source_id`, `GCharArray` (массив символов — строка) с именем `source_name`, еще один `GCharArray` с именем `source_url` и `GdkPixBuf` (массив пикселей) с именем `source_img`.

Что это было?

Последний тип — изображение; по умолчанию, его-то мы и будем отображать справа. Щелкните по объекту Дерево в древовидном списке объектов и выберите пункт меню *Edit*, чтобы редактировать свойства объекта более подробно в отдельном окне. Прокручивайте вниз, пока не увидите пункт «Столбец всплывающей подсказки», и введите значение «1». Этим мы выбираем первый столбец только что созданного списка (модели) в качестве всплывающей подсказки к каждой записи. Теперь перейдите на вкладку Иерархия и нажмите внизу списка кнопку *Добавить*, и добавьте одну колонку. Справа задается заголовок и набор свойств. Теперь щелкните на добавленном столбце правой кнопкой мыши и в появившемся меню выберите «Добавить дочерний элемент Изображение». В панели справа установите для пункта Объект зна-

чок (*Pixbuf*) в значение 3, или выберите `source_img` в выпадающем списке. Это позволит отображать наше изображения в списке в ячейках.

Перед выходом добавьте ссылку на обработчик сигнала. На вкладке Сигналы объекта Дерево, найдите пункт «row-activated» и добавьте запись с именем «row_chosen». Сохраните все!

Пришло время поработать с кодом. Прежде всего нам нужен хитрый метод для извлечения изображения-иконки для конкретной ленты. Мы можем указать URL и извлечь изображение *Pixbuf*. Легко! (Это если вы следили за нашей серией.)

```
def fetch_image(self,url):
    import feedparser,urlib
    #найдем имя изображения
    f=feedparser.parse(url)
    name = f.feed.image.href
    imgfile,data = urllib.urlretrieve(name)
    img=gtk.gdk.pixbuf_new_from_file(imgfile)
    img=img.scale_simple(150,150,gtk.gdk.INTERP_BILINEAR)
    return img
```

На досуге можно подумать и о кэшировании изображений (помните: имена файлов могут конфликтовать, так что не мешает добавлять к ним и хэш URL'a); но пока оставим это. Они, конечно, будут загружаться при каждом запросе, но благодаря их малому размеру вы едва ли почувствуете серьезное снижение скорости.

Объедините этот метод с другими в главном файле *MyApp.py*, и раз уж файл открыт, просмотрите код, следующий после комментария **# Code for other initialization actions should be added here** [остальной код инициализации разместить здесь]. Добавьте следующий код:

```
store1=self.builder.get_object("liststore1")
for index, sub in enumerate(self.preferences['subs']):
    # структура данных subs должна
    # содержать name и url
    img=self.fetch_image(sub[1])
    store1.append([index,sub[0],sub[1],img])
```

Переменная `store1` содержит объект `liststore`, генерируемый в UI, и мы можем заполнить его данными. Я так вижу, что данные о наших лентах будут храниться в настройках в виде списка: имя ленты и URL (мы сделаем это через минуту). Используя URL, мы добудем изображение и поместим весь набор в хранилище списка. Просмотр в цикле (по списку списков) создает дополнительную переменную — номер итерации; его мы использу-

Сохраняйте почтаще

Иногда нам случается напортачить. Для файлов исходных текстов это обычно не проблема, но для файлов интерфейса *Glade* практически фатально, потому что в них много необратимых действий.

По этой причине *Quickly* имеет встроенную систему управления версиями. Это не просто способ упаковки и публикации проекта на *Launchpad*; это полезная вещь по сути для всего. Если ввести команду **quickly save 'комментарий к новой версии'**, то новая версия вашего проекта занесется в систему контроля версий *Bazaar*. Проверить сохраненные версии можно, скомандовав **bzr log** — выведется список всех выполненных транзакций, вместе с заметками и датой-временем. Если угробилось вообще все, введите в главном каталоге приложения **bzr revert** — и файл с исходными текстами, графический интерфейс и любые другие файлы восстановятся из предыдущей версии.

Подсказка-бонус: при желании восстановить лишь часть проекта, выполните **quickly design** и **quickly edit**, укажите команду **revert** и сохраните файлы, которые хотите оставить как были.

Исходные тексты

Исходные тексты имеются на **LXF DVD**, но если вам лень их искать, у вас нет DVD-привода или вы читаете это в онлайн-версии, переведенной на русский, затем на польский и вновь на английский, то исходные тексты можно скачать напрямую с сайта **Ubuntu Launchpad**.

Посетите <http://code.launchpad.net/podofile> или воспользуйтесь **Bazaar** для повторного создания локального репозитория командой **bzr branch lp:podofile**. Архив содержит также файлы настройки **Quickly**, так что если вы уже установили **Quickly**, то можете поиграть с ним самостоятельно.

ем для создания идентификатора ID. Осталось только вернуться в файл **PreferencesMyAppDialog.py** и добавить следующее к переменным по умолчанию:

```
self._preferences["subs"]=[["Tux Radar podcast (ogg) "http://www.tuxradar.com/files/podcast/podcast_ogg.rss"]]
```

Поместите это сразу после умолчаний **savedir**, введенных нами ранее. Сохраните все опять и вновь запустите приложение. Теперь вы должны увидеть загруженный логотип **TuxRadar** в левой панели!

Создание приложений с графическим интерфейсом начинают напоминать упражнение в прокладке труб. В нашем первом списке необходимо что-то сделать, если кто-то выберет в нем элемент. Механизм действий такой: перехватываем сигнал, подаваемый списком, а затем делаем нечто с другим списком; надо лишь соединить между собой части кода. Нас не волнует, когда они будут исполняться – это дело пользователя; главное – продумать все последствия предпринимаемых действий.

Трубопроводы

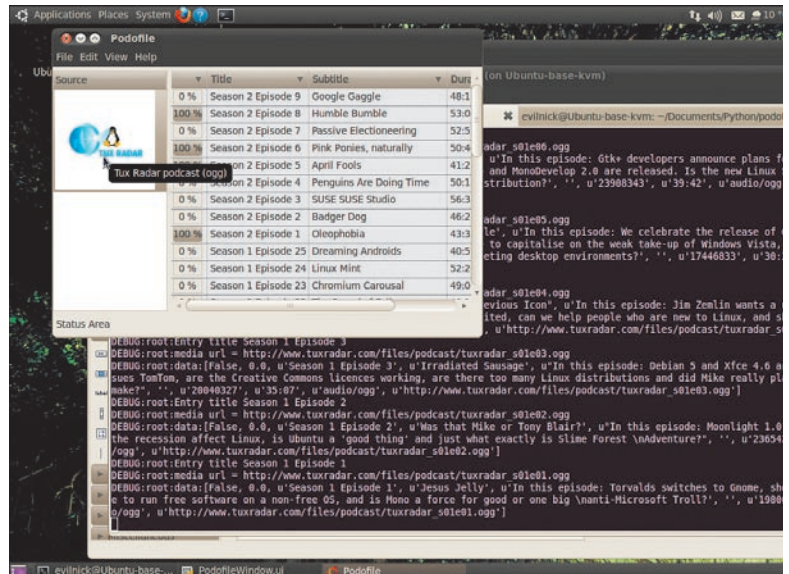
Но прежде чем подключаться ко второму списку, не худо узнать, что в нем содержится. Откройте **Glade** снова и разместите Дерево во втором прокручиваемом окне. В созданном для него представлении списке следует создать несколько столбцов – **boolean** (логическое) с именем **downloaded** [загружено], **gfloat** (число с плавающей точкой) с именем **progress** [прогресс] и восемь **gchararrays** – их я назвал **title** [название], **subtitle** [подзаголовок], **summary** [описание], **tags** [теги], **size** [размер], **duration** [длительность], **media_type** [тип содержимого] и **media_url** [URL содержимого]. Их назначение ясно из имен, и, честно говоря, дело до них скорее всего не дойдет.

Естественно было бы хранить все эти данные в объекте и копировать его в модель списка при необходимости, но в итоге у вас получится две копии всех данных, и забота об их синхронизации станет проблемой – накопив опыт, вы придете к выводу, что проще хранить данные в списке-представлении и ссылаться на него при необходимости, а не опрашивать переменные на каждом шагу. При добавлении столбцов в само Дерево, нет нужды добавлять все, однако первым добавьте **cellrendererprogress** и свяжите его с данными о прогрессе выполнения.

Для этого Дерева желательно убедиться, что в метке Общие для свойства **События** установлено значение **получать все**. Также добавьте сигнал с именем **file_chosen** в **row-activated**. Мы создадим для него метод обработки двойного щелчка на файле.

Для извлечения данных и содержимого в ваш список-представление добавьте такой код в главный файл:

```
def row_chosen(self,treeview,selection,treeviewcolumn):
    """ Обработчик сигнала для выбора элемента в строке источника """
    import feedparser
    listmodel=treeview.get_model()
    store2=self.builder.get_object("liststore2")
    store2.clear()
```



➤ Запуск приложения с опцией **-v** позволяет вести журнал. Я напудил их немало – они бесценны при отладке.

```
savedir=self.preferences["savedir"]
index=listmodel.get_iter(selection[0])
source_url=listmodel.get_value(index,2)
# получаем список файлов
f=feedparser.parse(source_url)
for entry in f.entries:
    media_url=entry.enclosures[0]["href"]
    logging.debug("media url = %s",media_url)
    fname=os.path.split(media_url)[-1]
    #проверяем что файл уже загружен
    downloaded=os.path.exists(os.path.join(savedir,fname))
    progress=0.0
    if downloaded:
        progress=100
    title= entry.title if entry.has_key("title") else "Unnamed"
    subtitle= entry.subtitle if entry.has_key("subtitle") else "no information"
    summary=entry.summary if entry.has_key("summary") else "no information"
    tags = ""
    if entry.has_key("tags"):
        for item in tags:
            tags+=item.term
    size= entry.enclosures[0].length
    duration= entry.itunes_duration if entry.has_key('itunes_duration') else "??:??"
    media_type= entry.enclosures[0].type
    logging.debug("data:%s",[downloaded,progress,title,subtitle,summary,tags,size,duration,media_type,media_url])
```

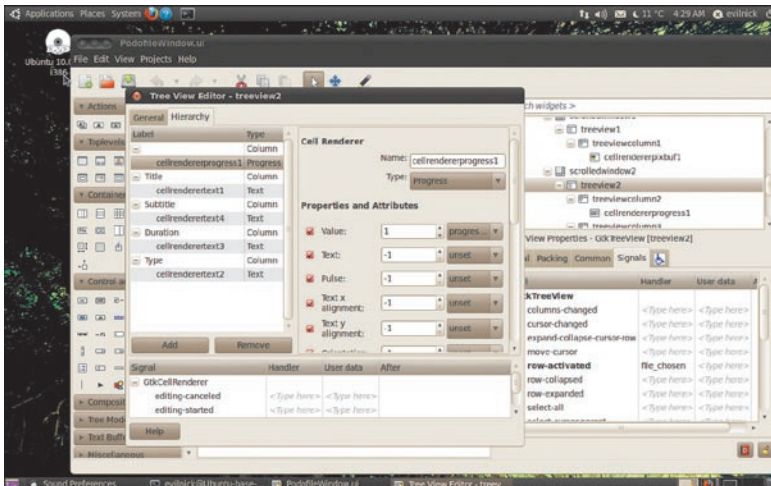
Пользуйтесь журналом

Древний прием напшиговывать программу операторами вывода различных значений и объектов для проверки – это отстой. Оно, конечно, помогает, но после отладки какой-либо части кода их нужно вычищать, не то ваш код разбухнет и при каждом запуске консоль будет заполняться мусором. Основной файл программы, создаваемый **Quickly**, инициализирует журнал, создаваемый соответ-

ствующим модулем Python, если вы запустите приложение с ключом **-v** (это можно делать и в **Quickly**). Так что для вывода сообщений просто используйте:

```
logging.debug("prefs= %s",self.preferences["subs"])
```

Этот метод следует стандартным правилам подстановки; его прелесть в том, что вывод он производит, только если его об этом просят.



Во втором дереве больше элементов. Не обязательно добавлять их только для отображения — можно, скажем, использовать текст описания для подсказки.

```
store2.append([downloaded,progress,title,subtitle,summary,
tags,size,duration,media_type,media_url])
```

Здесь нет ничего сверхсложного. Мы рассматривали чтение RSS-лент в предыдущих учебниках: при любой обработке RSS-лент, в формате Atom или нет, незачем искать добра от добра, а точнее — замечательного модуля **Feedparser** от Марка Пилгрима [Mark Pilgrim] (www.feedparser.org). По скрытым от человечества причинам этот модуль не включен в стандартный пантеон при установке приложения Ubuntu, но добавить его проще простого: **sudo apt-get install python-feedparser**. Подлинная красота этого модуля в том, что он прочесывает все мыслимые форматы лент и выдает приятный объект Python, с которым уже можно работать, не путаясь со всякими узлами, проверками используемой версии RSS и прочим. Передаете ему URL ленты — получаете четкую структуру. Для ленты подкастов возвращаемый объект будет содержать различные контейнеры `entry` [запись], с информацией о каждом доступном эпизоде. Они разбиваются на несколько стандартных полей — заголовок, описание, тэги, а также информацию о дате и URL для медиа-содержимого, что вполне достаточно для создания таблицы эпизодов.

Для выбора файла...

С этим разобрались; теперь обрабатываем сигнал, получаемый при выборе файла. Добавьте следующий метод:

```
def file_chosen(self, treeview, selection, column):
    import thread
    listmodel=treeview.get_model()
    index=listmodel.get_iter(selection[0])
    downloaded=listmodel.get_value(index,0)
    if downloaded:
        self.file_play(listmodel.get_value(index,9))
    else:
        self.file_download(listmodel,index)
    Он определяет, какой файл выбран, и проверяет, не загружен ли он. Если да, подставляется ссылка на другой обработчик:
    def file_play(self,url):
        """ Инициализирует PlayerDialog и проигрывает заданный файл """
        logging.debug("caught signal for file play")
        filename=os.path.split(url)[-1]
        filename=os.path.join(self.preferences["savedir"],filename)
        player=PlayerDialog.PlayerDialog()
        player.set_playfile(filename)
        player.run()
        player.destroy()
        Вообще-то здесь обращаются к новому диалогу. Чтобы все работало, добавьте диалог через Quickly, включив его в описание проекта. Для этого в каталоге проекта введите
        quickly add dialog Player
        Об остальном позаботится Quickly. Затем вы можете открыть эти файлы и изменить их при помощи quickly edit и quickly design. Наш интерфейс диалога — это всего лишь сигнал, показывающий: что-то случилось, и мы отреагировали. Если честно, то и в коде проигрывателя мало что происходит. Это просто быстрый способ издать звук.
        В конце метода finish_initializing нужно создать проигрыватель из модуля gst (ах да, необходимо добавить в начало файла import gst):
        self.player=gst.element_factory_make("playbin","PodPlayer")
        Затем мы просто реализуем метод проигрывания файла:
        def set_playfile(self,filename):
            #получает путь к файлу и запускает проигрыватель
            uri ="file://"+filename
            print uri
            self.player.set_property("uri",uri)
            self.player.set_state(gst.STATE_PLAYING)
```

Да не забудьте остановить проигрыватель. Не обольщайтесь, что объект будет уничтожен вместе с модальным диалогом: звук

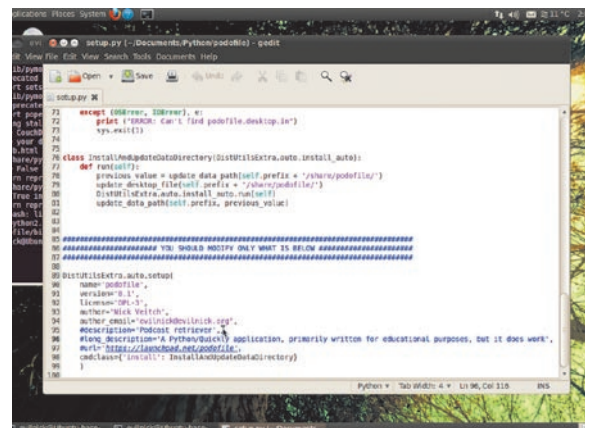
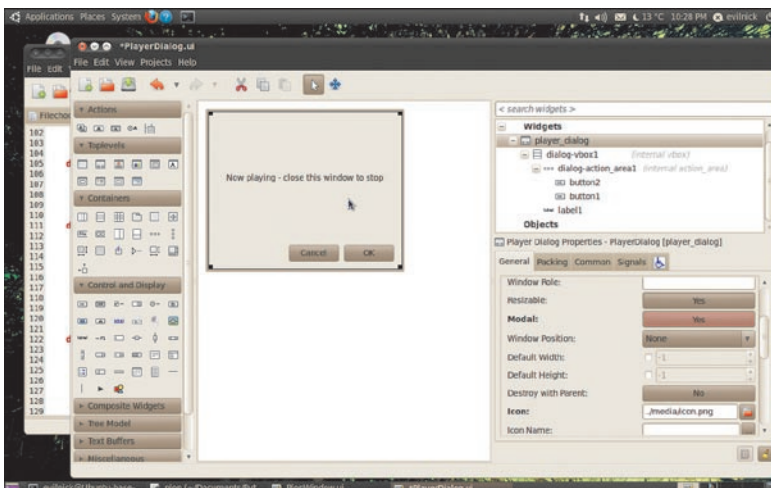
Об остальном позаботится *Quickly*. Затем вы можете открыть эти файлы и изменить их при помощи *quickly edit* и *quickly design*. Наш интерфейс диалога — это всего лишь сигнал, показывающий: что-то случилось, и мы отреагировали. Если честно, то и в коде проигрывателя мало что происходит. Это просто быстрый способ издать звук.

В конце метода **finish_initializing** нужно создать проигрыватель из модуля **gst** (ах да, необходимо добавить в начало файла **import gst**):

```
self.player=gst.element_factory_make("playbin","PodPlayer")
Затем мы просто реализуем метод проигрывания файла:
def set_playfile(self,filename):
    #получает путь к файлу и запускает проигрыватель
    uri ="file://"+filename
    print uri
    self.player.set_property("uri",uri)
    self.player.set_state(gst.STATE_PLAYING)
```

Да не забудьте остановить проигрыватель. Не обольщайтесь, что объект будет уничтожен вместе с модальным диалогом: звук

Как видно, проигрыватель — лишь модальный диалог, порождающий экземпляр *Gstreamer*, но сторонние разработки приветствуются!



Перед созданием пакета измените `setup.py` и добавьте в файл все элементы в указанных местах.

» Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.

в Linux – штука капризная, и вы неоднократно можете столкнуться с ситуацией упорного проигрывания MP3-файлов, хотя предполагалось, что воспроизведение остановлено. Можно просто поместить вызов остановки проигрывателя в обработчик кнопок OK и Cancel.

```
self.player.set_state(gst.STATE_PAUSED)
```

Этот код позаботится о том, чтобы подкасты не перетрудились.

Осталось позаботиться о самой загрузке медиа-файлов. Для получения файлов воспользуемся `urllib.urlretrieve()`: это часть стандартного пакета Python, и она работает достаточно хорошо. В этом коде есть два хитрых трюка, и его следует немного пояснить.

```
def file_download(self, listmodel, index):
    import urllib, thread
    url=listmodel.get_value(index,9)
    savename=os.path.split(url)[-1]
    savename=os.path.join(self.preferences["savedir"],savename)
    reporthook = lambda a,b,c : self.
progress_update(listmodel,index,a,b,c)
    thread.start_new_thread( urllib.urlretrieve,(url,savename,rep
orthook))
```

Метод `urlretrieve` предусматривает обратный вызов. Беда в том, что последний возвращает лишь данные о загруженных блоках, и ничего о загружаемом файле. Запустив сразу несколько загрузок, мы уже не сможем рассматривать их по отдельности. Я в общем против применения лямбда-функций (функций, встроенных в Python) – они нередко усложняют понимание кода. Однако в данном случае использование лямбда-функций в качестве прокси означает, что мы можем добавлять полезную информацию к отклику процесса загрузки, и передавать ее в следующую функцию. Прежде чем браться за это, рассмотрим еще одну используемую здесь уловку – а именно, потоки.

Приятные потоки

Простой случай потоков (см. последнюю строку) означает, что можно запускать поток, представив функцию и ее аргументы в виде кортежа. Без потоков работа приложения будет приостановлена до тех пор, пока файл не загрузится – оно не будет ни на что откликаться, и пользователь может решить, что оно зависло. Вдобавок наш удобный индикатор прогресса будет бесполезен. Для создания обратной связи нам необходим второй метод:

```
def progress_update(self, listmodel, index, a, b, c):
    """ Функция обратного вызова, отражающая
    прогресс в таблице. Необходимы потоки! """
    blocks = float(c/b)
    progress=float(a/blocks)*100
    if progress > 100:
        progress = 100
    listmodel.set(index,0,True)
    listmodel.set(index,1,progress)
```

Здесь вычисляется процент завершенности и соответственно обновляется модель представления списка. Можно запустить несколько загрузок одновременно, и приложение будет обновлять их все. Исходная идея была в том, чтобы индикатор прогресса отображался под каждым элементом на всю ширину окна. К сожалению, `GTK` не поддерживает такого напрямую (по крайней мере, пока), но если вы желаете создать ваш собственный вариант отрисовки, то сделайте это и дайте нам знать, пожалуйста!

Чтобы заставить потоки работать, припишите следующее в конец основного файла, сразу после `import optparse`:

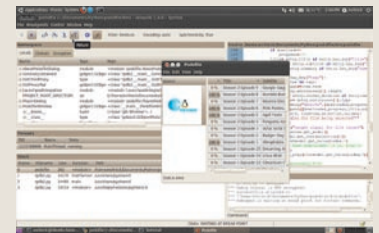
```
import gobject
gobject.threads_init()
```

Если этого не сделать, `GTK`-часть приложения не поладит с модулем потоков Python.

Отладка в Quickly

Даже маститые кодеры здесь у нас в Башне `LXF` иногда ошибаются в суждениях. Вообще-то люди, имеющие привычку все оценивать, считают, что 80% времени разработки ПО уходит на исправление ошибок. Но исправление ошибок – вовсе не каторга: это путь к открытиям (как мы пытаемся убедить сами себя), дающий нам ощущение востребованности и полезности.

Для Python имеется несколько прекрасных инструментов отладки, но `Quickly` поставляется с одним уже настроенным и готовым к работе, так что мы воспользуемся им! Возможно, ранее вы не сталкивались с `Winpdb`, а он содержит все функции для пошагового перемещения между точками останова, анализа пространства имен и проверки



Это не центр управления энергоподстанцией, это окружение для отладки.

поведения переменных. Он определенно быстрее стандартного `PDB`, и узнать о нем больше можно на <http://winpdb.org/docs>.

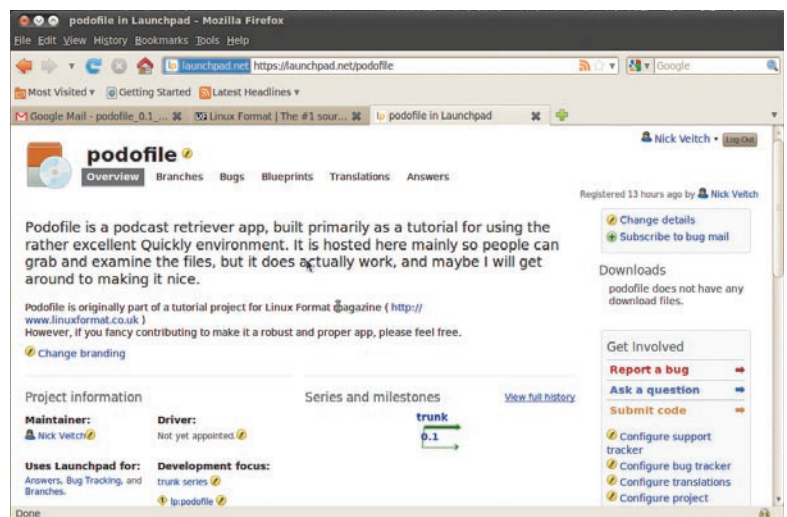
Летите на волю

Если вы пробовали приложение по ходу урока, то теперь увидите, что оно вполне готово к употреблению. Конечно, доработать еще есть что, и, кстати, в исходном коде в сети или на DVD разбросано множество примечаний и задач [TODO]. Для начала, несколько пунктов меню не работают! Ох, и еще одна загвоздка – мы так и не дали пользователю возможность добавлять свои ленты подкастов (это если им мало одного `TuxRadar...`).

Однако мы должны выполнить последнюю часть нашей миссии, то есть выпустить приложение на волю. Командами `Quickly` это делается легко, нужна только учетная запись на `Launchpad`. Если у вас такой нет, создать ее недолго; но чтобы выгрузить данные в `PPA`, потребуется также иметь ключ `GPG` и открытый ключ `SSH`, для проверки ваших файлов.

Ну, а для простого создания `Deb`-файла и того не надо. Просто введите команду `quickly package`, и скрипты потрудятся за вас, создав готовый к распространению `Deb`-пакет. Если вы упорно желаете выгрузиться на `Launchpad`, сделайте это командой `quickly release`, просто следуя ее инструкциям. Однако хорошей идеей будет сначала представиться `Bazaar`, а затем убедиться, что ваш почтовый адрес соответствует учетной записи `Launchpad`:

```
bzr whoami 'My Name <email@example.com>'
Quickly размещается на Launchpad (https://launchpad.net/quickly); создав свою учетную запись, вы сможете заходить, задавать вопросы и тусоваться с крутыми кодерами. LXF
```



➤ Если вы планируете опубликовать свой код на `Launchpad`, имеет смысл сперва зарегистрировать для него `PPA`-слот, а затем вызвать ваш проект `Quickly` с этим именем.

Perl: Создаем

Часть 1: Язык, созданный для обработки текстов, может неплохо справляться и с графикой. **Михаил Смирнов** объяснит, что к чему.



Наш эксперт

Михаил Смирнов
Независимый web- и прикладной программист в области цифровой обработки изображений, голографии и распознавания образов.

Как известно, язык программирования Perl идеально приспособлен для многочисленных приложений по обработке текста. Однако возможности Perl этим не исчерпываются. Применение графических модулей в Perl позволит вам быстро рисовать изображения, состоящие из линий, окружностей и текста, манипулировать цветом, вырезать и вставлять другие изображения, выполнять заливку, сохранять на диске файлы изображений в различных форматах, манипулировать видеоданными этих файлов. Наибольший интерес представляет использование графических модулей Perl в web-приложениях. Графическая библиотека GD имеет интуитивно понятный интерфейс, позволяющий быстро и компактно реализовать пользовательский код программ на Perl даже начинающим программистам.

Эффективность использования графического модуля GD объясняется просто: библиотека GD написана на языке C и совместима со всеми основными платформами (Linux, Windows, ...). Интерфейс между Perl и библиотекой GD обеспечивает модуль GD.pm, который использует GD как базис для создания графических классов в Perl. GD.pm является самозагружаемым интерфейсным модулем, с помощью которого можно создавать изображения на лету, модифицировать существующие файлы изображений и синтезировать новые.

Привет, GD!

Библиотека GD позволяет создавать цветные рисунки, используя большое количество графических примитивов, и выводить

их в различных графических форматах. Модуль GD определяет три основных класса:

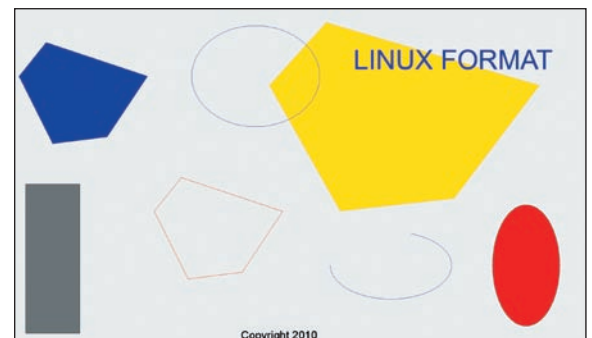
- » класс изображений **GD::Image**, который предназначен для захвата видеоданных изображения и выполняет вызов методов графических примитивов;
- » класс шрифтов **GD::Font**, который берет статические шрифты и использует их для воспроизведения текста в графическом виде;
- » класс примитивов полигона **GD::Polygon**, применяемый для сохранения списка вершин многоугольников при рисовании сторон полигона и обеспечивающий их воспроизведение в виде изображения.

Обобщенный подход использования графического интерфейса GD выглядит следующим образом. На первом этапе, при вызове (инициализации) метода `GD::Image::new(Wx,Hу)`, создается новое пустое изображение, ширина и высота которого задаются параметрами `Wx` и `Hу`. В результате вызова метода будут возвращены видеоданные изображения. Другие методы класса позволяют инициализировать изображение из уже существующих файлов, которые необходимо предварительно прочитать с диска.

На втором этапе, при вызове метода `colorAllocate()` выполняется добавление цвета к цветовой таблице изображения. Входными параметрами вызова является тройка основных цветов: красный, зеленый и синий, которые задаются целыми числами в диапазоне значений от 0 до 255. Метод возвращает индекс цвета цветовой таблицы изображения.

После этого, на третьем этапе, можно сделать некоторый рисунок, используя набор графических примитивов класса `GD::Image` и набор методов класса `GD::Polygon`. Полигоны (многоугольники) создаются с помощью метода `new()`, а добавление новых вершин полигона возложено на метод `addPt()`.

И, наконец, когда рисунок готов, вы можете преобразовать его, например, в графический формат GIF, с помощью метода `gif()`. Метод будет возвращать двоичное содержимое изображения в формате GIF. Обычно полученный результат выводится в браузер или сохраняется на диске в виде GIF-файла. В качестве примера приведем скрипт, который рисует простые фигуры, манипулирует ими самими и их цветом, а также рисует текст с помощью статических шрифтов:



» Рис. 1. Пример рисования простых объектов.

Изображения

```
#!/perl/bin/perl
use GD;
$im = new GD::Image(425,250); #--Инициализация изображения
$black = $im->colorAllocate(0, 0, 0); #--Создаем цветовую
палитру
$red = $im->colorAllocate(255, 0, 0);
$blue = $im->colorAllocate(0,0,255);
$yellow = $im->colorAllocate(255,250,0);
$bg = $im->colorAllocate(240,240,240);
$gray = $im->colorAllocate(128,128,128);
$im->fill(0,0,$bg); #--Делаем заливку фона
$im->rectangle(0,0,424,249,$black); #--Рисуем границы
изображения
$poly = new GD::Polygon; #--Инициализация полигона
$poly->addPt(5,50);
$poly->addPt(25,25);
$poly->addPt(100,50);
$poly->addPt(70,95);
$poly->addPt(30,100);
$im->filledPolygon($poly,$blue); #--Создание полигона
$poly->offset(100,100); #--Смещение полигона
$im->polygon($poly,$red); #--Рисуем только контур
$poly->map($poly->bounds,190,10,390,150); #--Изменение
масштаба полигона в 2 раза
$im->filledPolygon($poly,$yellow);
$im->arc(180,50,95,75,0,360,$blue); #--Рисуем эллипсы и дуги
$im->arc(280,190,90,50,290,180,$blue);
$im->arc(380,190,50,90,0,360,$blue);
$im->fill(380,190,$red);
$im->filledRectangle(10, 130, 50, 240, $gray); #--Рисуем
четырёхугольник
$im->rectangle(10,130,50,240,$black);
$string="LINUX FORMAT";
$im->string(gdMediumBoldFont,260,25,$string,$blue); #--Задаем
статические шрифты
$im->string(gdSmallFont,170,235,"Copyright 2010",$black);
print "Content-Type: image/gif\n\n";
binmode STDOUT;
print $im->gif(); #--Преобразование данных в формат GIF
и вывод
```

Показанный выше код позволяет web-программисту познакомиться на практике с основными командами для рисования и может послужить основой для самостоятельного создания более серьезных программ. Полученный результат можно посмотреть на рис. 1.

Основные методы

В иерархии интерфейсов GD наибольшее распространение получили два типа версий для работы с растровой графикой. Первый тип поддерживает манипулирование и преобразование изображений в формате GIF. Второй тип интерфейсов GD позволяет работать с такими популярными графическими форматами, как PNG и JPEG.

Для закрепления полученных выше практических навыков рассмотрим несколько «теоретических» моментов инициализации основных методов GD. Вызов метода **GD::Image::new(Wx,Hx)** возвращает новое пустое изображение, например:

```
$myImage = new GD::Image(100,100) || die;
```

Метод создает пустой бланк изображения размером 100×100 точек. Если размер опущен, он будет составлять 64×64 точек. Метод **GD::Image::newFromGif(FILE)** создает изображение на основе данных файла в формате GIF, предварительно прочитанного с диска. Параметр FILE представляет собой указатель файла (дескриптор). Дескриптор файла указывается как один из параметров функции `open()`. Если открытие файла было успешным, то вызов возвращает инициализированное изображение, которое затем может быть подвергнуто различным манипуляциям. Например:

```
open (MyGIF,"images/ballon.gif") || die;
$myImage = newFromGif GD::Image(MyGIF) || die;
close MyGIF;
```

MyGIF здесь – указатель файла. Вызов метода **GD::Image::gif** возвращает видеоданные изображения в формате GIF. Затем их можно вывести в браузер и/или записать в файл. Например:

```
open (MyGIF,">images/ballon.gif");
binmode(MyGIF);
print MyGIF $im->gif;
close (MyGIF);
print "Content-Type: image/gif\n\n";
binmode STDOUT;
print $im->gif;
```

Метод **GD::Image::colorAllocate(R,G,B)** предназначен для задания и управления цветом с помощью RGB-спецификации – цветовых компонент R (красный), G (зеленый) и B (синий), и в результате возвращает индекс цветовой таблицы. Пример задания желтого цвета:

```
$yellow = $myImage->colorAllocate(255,255,0);
```

Полученные индексы впоследствии можно использовать для задания общего фона изображения или для заливки фигур определенным цветом. Метод **GD::Image::getPixel(x,y)** возвращает ин-

»



» Рис. 2. Пример синтеза изображения.

» Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.

декс цветовой таблицы под определенной точкой изображения с координатами (x,y). Вызов этого метода может быть скобинирован с функцией rgb(), если вам потребуется восстановить список RGB-компонент под выбранной точкой. Например:

```
$index = $myImage->getPixel(20,100);
($r,$g,$b) = $myImage->rgb($index);
```

Метод **GD::Image::transparent(colorIndex)** отмечает индекс **colorIndex** как прозрачный. Участки изображения, нарисованные этим цветом, будут невидимы. Метод применяется для создания прозрачных фонов изображений в Web. Прозрачным в изображении может быть только один цвет. Ниже, например, им будет белый:

```
open(myGIF,"transptest.gif");
$im = newFromGif GD::Image(myGIF);
$white = $im->colorClosest(255,255,255);
$im->transparent($white);
print $im->gif;
close myGIF;
```

Команды рисования

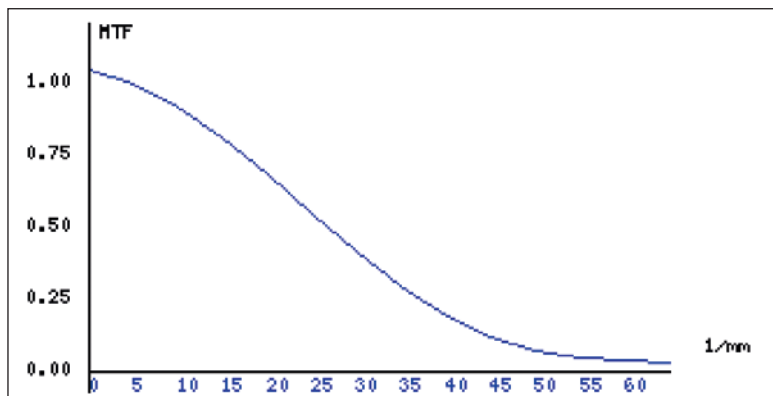
Эти команды хотя и не занимают в иерархии методов GD особого положения, но выделяются своей выраженной функциональностью. Интуитивная простота команд (методов) позволит вам быстро нарисовать целый ряд примитивов или запрограммировать произвольный рисунок. Рассмотрим ряд наиболее употребляемых методов.

Метод **GD::Image::setPixel(x,y,color)** создает точку с координатами (x,y) с определенным цветовым индексом color. Он ничего не возвращает. Система координат начинается в верхнем левом углу и заканчивается в нижнем правом. Нетрудно представить, что растровый перебор координат при наличии функциональной зависимости цветового индекса color от координат (x,y) может использоваться для синтеза изображений. Реализуем пример кода для формирования двумерного распределения функции $w=\sin(x,y)$:

```
$im = new GD::Image($X,$Y);
$arg=$period*4*$pi*$pi;
for($j=0;$j<$Y;$j++){
    $y = ($j-$Y/2)/$Y;
    $phy = $y*$y;
    for($i=0;$i<$X;$i++){
        $x = ($i-$X/2)/$X;
        $phx = $x*$x;
        $w = int($a0 + $amp*sin($arg*($pi - $phx - $phy)));
        $gray = $im->colorResolve($w,$w,$w);
        $im->setPixel($i,$j,$gray);
    }
}
```

Метод **colorResolve(r,g,b)** возвращает индекс цвета, который точно соответствует указанным красным, зеленым и синим компонентам. Если такой цвет не находится в цветной таблице, то метод добавляет цвет в таблицу и возвращает его

➤ Рис. 3. Пример построения графика функции одной переменной.



индекс. На рис. 2 представлен результат синтеза изображения, созданного в соответствии выше приведенным кодом.

Метод **GD::Image::rectangle(x1,y1,x2,y2,color)** рисует четырехугольник, стороны которого имеют определенный цвет color. Координаты (x1,y1) и (x2,y2) являются верхним левым и правым нижним углами, соответственно. Пример рисования квадрата размером 90x90 точек:

```
$myImage->rectangle(10,10,100,100,$blue);
```

Метод **GD::Image::polygon(polygon,color)** рисует многоугольник с заданным цветом color, при этом число вершин не должно быть менее трех. Если последняя вершина полигона не создана, метод будет закрыт. Пример рисования треугольника:

```
$poly = new GD::Polygon;
$poly->addPt(50,0);
$poly->addPt(99,99);
$poly->addPt(0,99);
$myImage->polygon($poly,$blue);
```

Метод **GD::Image::line(x1,y1,x2,y2,color)** рисует линию с определенным цветом color от точки с координатами (x1,y1) до точки с координатами (x2,y2). Пример рисования диагональной синей линии от точки с координатами (7,7) до точки с координатами (154,154):

```
$myImage->line(7,7,154,154,$blue);
```

Метод рисования **line()** очень часто применяется для построения графиков функции одной переменной. Фрагмент кода скрипта, реализующий основной цикл формирования линии графика, а также оси абсцисс и значений переменной по оси абсцисс, представлен ниже:

```
$f=0;
for($i=0;$i<$N-1;$i++){
    $im->line($x,$y[$i],$x+$dx,$y[$i+1],$blue);
    $im->line($x,$yh,$x+$dx,$yh,$black);
    $im->string(gdTinyFont, $x, $yh+2, $f, $blue);
    $f +=5;
    $x=$x+$dx;
}
}
```

Пример рисования графика функции одной переменной показан на рис. 3.

С помощью метода **GD::Image::arc(cx,cy,Wx,Hy,start,end,color)** можно рисовать дуги и эллипсы. Координаты (cx,cy) являются центром эллипса (дуги), а параметры (Wx,Hy) являются шириной и высотой эллипса, соответственно. Часть эллипса покрывается дугой, длина которой управляется параметрами start и end. Эти параметры задаются в угловой мере от 0 до 360 градусов. Пример рисования полного эллипса:

```
$myImage->arc(100,100,50,35,0,360,$blue);
```

Метод **GD::Image::fill(x,y,color)** служит для заливки областей изображения с заданным цветом color. Цвет будет распространяться по изображению, начиная от точки (x,y) до границ объекта, на которых появляется разностный цвет. Пример заливки синим цветом, начиная с точки с координатами x=50 и y=40:

```
$myImage->rectangle(10,10,100,100,$black);
$myImage->fill(50,40,$blue);
```

Копирование

Команды копирования изображений выделяются своей функциональностью и важны с точки зрения создания эффективного кода при рисовании множества повторяющихся участков изображения. При написании кода программы этот метод позволит запомнить некоторую область изображения аналогично тому, как это делается с обычными массивами данных в Perl.

Метод **GD::Image::copy(sourceImage,dstX,dstY,srcX,srcY,Wx,Hy)** предназначен для копирования прямоугольных областей от одного изображения к другому. Изображение-источник задается с помощью параметра sourceImage. Параметры (srcX,srcY) задают верхний левый угол прямоугольника в изображении-источнике,



► Рис. 4. Пример слияния двух изображений с эффектом прозрачности.

а (*Wx*, *Hx*) указывают ширину и высоту области копирования. Параметры (*dstX*,*dstY*) управляют координатами точки, в которой будет начинаться наложение копии на изображение-получатель. Вызов метода возвращает данные изображения-получателя. Пример копирования области *\$srcImage* размером 25×25 точек в область *\$myImage* с началом в точке (10,10) показан ниже:

```
$myImage = new GD::Image(100,100);
$srcImage = new GD::Image(50,50);
$myImage->copy($srcImage,10,10,0,0,25,25);
```

С изобразительной точки зрения, более интересным методом является метод **GD::Image::copyMerge(*sourceImage*,*dstX*,*dstY*,*srcX*,*srcY*,*Wx*,*Hx*,*opaque*)**, который отличается от предыдущего всего лишь одним параметром *opaque*. Он может пригодиться для создания коллажей. С помощью параметра *opaque* в диапазоне значений от 0 до 100 задается степень прозрачности одного изображения относительно другого, например:

```
$image1->copyMerge($image2,$left,$top,0,0,$wx,$hy, 60);
```

На рис. 4 показан пример слияния с эффектом прозрачности двух изображений одного размера. При этом прозрачность второго изображения относительно первого составляет 60 %.

Выводим текст

Интерес web-программистов к рисованию текста связан со множеством задач, возникающих при создании графических элементов web-страниц: от случайного набора символов в задаче авторизации пользователя (создание так называемой *Captcha*), до рисования водяных знаков в изображениях.

Интерфейс GD поддерживает минимальный набор статических шрифтов и позволяет рисовать символы и строки текста в графическом виде. Вывод может выполняться как в горизонтальном направлении, так и в вертикальном. Доступными для рисования шрифтами являются следующие: **TinyFont**, **SmallFont**, **Medium-BoldFont**, **LargeFont** и **GiantFont**. Каждый из этих шрифтов может быть импортирован как глобальная константа или как пакет объектов класса **GD::Font**, например: **GD::Font::gdTinyFont**.

Непосредственно для рисования применяется два основных метода: **string()** и **stringUp()**. Метод **string(*font*,*x*,*y*,*string*,*color*)** рисует строку текста, начиная с позиции (*x*,*y*), в заданном цвете и выбранном шрифте. Метод **stringUp(*font*,*x*,*y*,*string*,*color*)** рисует текст с поворотом на 90 градусов в направлении, противоположном вращению часовой стрелки.

Рассмотренные выше шрифты имеют весьма ограниченный диапазон размеров, и при этом только средний шрифт позволяет выделять символы как жирные. Другой существенный недостаток

статических шрифтов – это ограничение по локализации исключительно латиницей.

Использование динамических шрифтов TrueType обеспечивает при рисовании текста практически неограниченные возможности, и прежде всего, локализацию под любой тип кодировки, применяемый в web-программировании. Метод для вызова шрифтов имеет вид **GD::Image->stringTTF()**. Разнообразие же существующих шрифтов снимает ограничения на их доступность. Один из вариантов вызова метода **stringTTF()** имеет вид:

```
$image->stringTTF($fgcolor,$fontname,$ptsize,$angle,$x,$y,$str),
```

где аргументы команды – это:

- » индекс цвета *fgcolor* для рисования строки текста,
- » путь *fontname* к выбранному файлу шрифта (*.ttf*),
- » желаемый размер *ptsize* шрифта,
- » угол поворота *angle* в радианах,
- » декартовы координаты *x*,*y* начала рисования,
- » строка текста *str*.

На рис. 5 показан результат рисования строки текста несколькими кириллическими шрифтами: *buaril.ttf*, *bukursv.ttf*, *buroman.ttf*, *butimit.ttf*, *pushkin.ttf*, *butimes.ttf*.

Форматы PNG и JPEG

Как было отмечено выше, второй тип интерфейсов GD предоставляет возможность использовать графические форматы PNG и JPEG. Для работы с графическим форматом PNG применяется метод **GD::Image->newFromPng(*\$file*)**, который с помощью дескриптора файла может создавать изображения в этом формате. Если открытие файла было успешным, то вызов возвращает инициализированное изображение, например:

```
open (myPNG,"test.png") || die;
$myImage = newFromPng GD::Image(*myPNG) || die;
close myPNG;
print "Content-type: image/png\n\n";
binmode STDOUT;
print $im->png();
```

Существует другой вариант вызова с непосредственным указанием файла изображения:

```
$myImage = newFromPng GD::Image('test.png')
```

Для работы с форматом JPEG применяется метод **GD::Image->newFromJpeg(*\$file*)**. Этот метод будет создавать изображение в формате JPEG после чтения файла с диска в том же формате:

```
$quality=75;
$image = GD::Image->newFromJpeg($file);
print "Content-type: image/jpeg\n\n";
binmode STDOUT;
print $image->jpeg($quality);
```

Отличительная особенность метода **jpeg()** – наличие параметра качества *\$quality*, задаваемого как целое в диапазоне значений от 0 до 100. Чем больше его значение, тем выше качество изображения, и, естественно, больше размер файла на диске. Если он не задан, по умолчанию будет выбрано среднее качество. **1x39**

```
buaril.ttf: абвгдежзийАБВГДЕЖЗИЙ 012345
bukursv.ttf: абвгдежзийАБВГДЕЖЗИЙ 012345
buroman.ttf: абвгдежзийАБВГДЕЖЗИЙ 012345
butimit.ttf: абвгдежзийАБВГДЕЖЗИЙ 012345
pushkin.ttf: абвгдежзийАБВГДЕЖЗИЙ 012345
butimes.ttf: абвгдежзийАБВГДЕЖЗИЙ 012345
```

► Рис. 5. Пример рисования текста динамическими шрифтами.

» **Через месяц** Оценим разрешающую способность и повысим качество изображений.

OOo Calc:

После того, как функция написана и работает правильно, надо добиться, чтобы она работала быстро. **Александр Маджугин** знает, как этого добиться.

В прошлых статьях мы научились создавать пользовательские функции для *Calc* и делать их работу стабильной и удобной. Но при любых сложных вычислениях есть ещё один важный аспект – это производительность. Если вы создаёте пользовательские функции, ваши вычисления уже можно назвать сложными, так как для простых в большинстве случаев хватает и встроенных функций электронных таблиц.

В этой статье мы начнём говорить об оптимизации вычислений в *Calc*, затрагивая вопросы оптимизации кода в функциях Basic, применительно к реализации, встроенной в *OpenOffice.org*. Статья окажется полезной не только тем, кто создаёт пользовательские функции *Calc*, но и всем, кто занимается разработкой макросов для *OOo*.

Датчик скорости

Измерять производительность можно двумя способами: по времени вычисления фиксированного количества результатов и по количеству результатов, полученных за фиксированное время. И тот, и другой имеют полное право на существование, но первый обычно удобнее в реализации. Второй же лучше подойдёт для создания тестов производительности.

Посмотрим на следующий код:

```
Sub Chronometr
  Dim IStartTime As Long ' Время запуска
  Dim IResult As Long ' Время работы
  IStartTime = getSystemTicks
  MyFunction
  IResult = getSystemTicks - IStartTime
  Print IResult
End Sub
```

Он измеряет время однократного вычисления функции *MyFunction* путём нахождения разности значений счётчика *getSystemTicks* до вызова функции и после его завершения.

Как правило, одного вычисления недостаточно – какими бы медленными ни были интерпретаторы, скорость всё равно достаточно высока, и шанс, что результатом замера будет 0, весьма велик. Поэтому практически всегда приходится использовать цикл с многократным вычислением функции: чем он длиннее, тем меньше погрешность и накладные издержки измерения. На практике замер должен длиться не меньше 8–10 секунд.

Рассмотрим практический пример, а нашим подопытным станет функция *FXOR* из предыдущей статьи – Листинг 1. Для измерения скорости её работы преобразуем процедуру *Chronometr* таким образом:

```
Sub Chronometr
  Dim IStartTime As Long ' Время запуска
  Dim IResult As Long ' Время работы
  Dim IFuncResult As Long ' Переменная для приёма
  результата
  Dim I As Long
```

```
IStartTime = getSystemTicks
For I = 1 To 100000
  IFuncResult = FXOR(255,I)
Next
IResult = getSystemTicks - IStartTime
Print IResult
End Sub
```

Мне понадобилось 100 000 шагов цикла для того, чтобы время работы на моём ноутбуке приблизилось к 10 секундам. Если у вас производительная машина, вам может потребоваться больше. Для приёма результата я создал специальную переменную *IFuncResult*: это не обязательно, но я люблю, чтобы всё было «по-честному». В качестве второго аргумента *FXOR* я использовал счётчик цикла – это удобно, особенно если время работы вашей функции может зависеть от значения аргументов.

Для повышения честности замера иногда бывает полезно вычислить время холостого хода тестовой процедуры. Просто закоментируйте строку *IFuncResult = FXOR(255,I)*, выполните несколько запусков, и, запомнив время работы процедуры, вычитите его из конечного результата, либо добавьте в начало функции следующий код:

```
IStartTime = getSystemTicks
For I = 1 To 100000
Next
IResult = getSystemTicks - IStartTime
```

Строку *IResult = getSystemTicks - IStartTime* следует заменить на *IResult = getSystemTicks - IStartTime - IResult*. Тогда холостой ход будет вычитаться из результата автоматически. Другим способом снижения накладных издержек работы цикла является многократный вызов процедуры внутри него.

Для перегруженных функций важным является ещё и тип используемых в тесте аргументов. Например, вызов *FXOR* только для целых дал у меня результат около 10 000 миллисекунд, а вызов её же для строк длиной 6 символов дал уже в среднем 47 000 миллисекунд. Старайтесь замерять общую производительность функции в условиях, близких к реальным. Например, если предполагается, что *FXOR* будет использоваться для целых примерно в 75% случаев, а для строк средней длиной 6 символов только в 25% случаев, то код внутри цикла должен выглядеть примерно так:

```
IFuncResult = FXOR(65535,I)
IFuncResult = FXOR(0,I)
IFuncResult = FXOR(I,I)
sFuncResult = FXOR("djkelj","saljcx")
```

Эти рекомендации касаются только общей производительности. При оптимизации конкретного участка кода, напротив, следует снизить все издержки на вычисления вне его. Для достижения данной цели используйте знания о структуре кода и конструкции своего алгоритма. Например, при оптимизации обработки строк в нашем примере вообще не стоит вызывать



ОПТИМИЗАЦИЯ

саму функцию FXOR, дабы избежать лишних издержек на перегрузку, которые постоянны. Следует вызывать сразу её подфункцию SXOR. В данном случае это помогает снизить время работы теста почти на 20%.

Оптимизация кода

Перейдем непосредственно к оптимизации, и начнем с подфункций. Внимательно посмотрим на SXOR. Можно ли здесь что-то улучшить? С первого взгляда – вроде бы нет. Но вот что выйдет, если мы применим лишь пару из самых важных правил оптимизации:

» Общее решение всегда медленнее решения для частного случая. (Если вы считаете, что такой стиль – дурной тон, то дочитайте, пожалуйста, статью до конца – я вернусь к этому вопросу).

» В первую очередь следует минимизировать количество вычислений внутри цикла.

Какие мы можем выделить частные случаи? Во-первых, когда второй аргумент равен по длине или длиннее первого. В этом случае становится лишней операция деления значения счётчика по модулю длины второго аргумента – он и так не выйдет за пределы строки. Соответственно, вычисление очередного символа результата внутри цикла можно переписать так: **sResult = sResult & Chr(ASC(Mid(sFirst,I,1)) XOR ASC(Mid(sSecond,I,1)))**, минимизировав число операций внутри цикла.

Во-вторых, длина второго аргумента может быть равна одному символу. Здесь вообще становятся не нужны функции Mid и ASC внутри цикла, и операцию нахождения кода символа второй строки можно вынести:

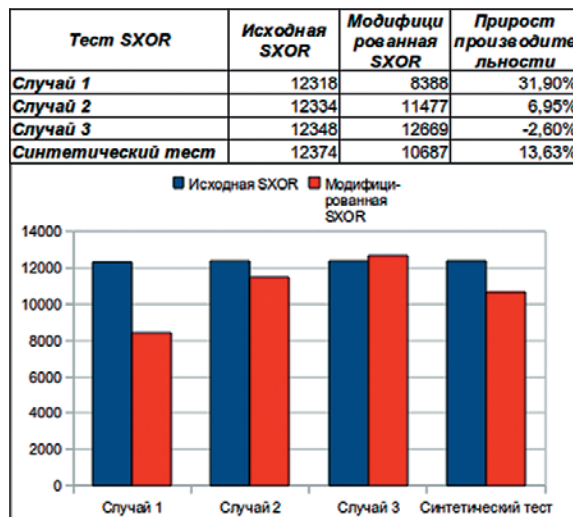
```
iSC = ASC(sSecond)
For I = 1 To ILF
    sResult = sResult & Chr(ASC(Mid(sFirst,I,1)) XOR iSC)
Next
```

Ну и третий случай – когда второй аргумент короче первого, но длиннее одного символа. Здесь код останется неизменным. В итоге функция примет вид, приведенный в Листинге 2.

Как видно, функция стала длиннее. В неё добавились новые операции сравнения. Ну, а что же с производительностью? Согласно рис. 1, модификация принесла нам более чем 13% рост производительности в синтетическом тесте (по 5000 расчётов для наборов с длиной второго аргумента 1, 3 и 16 символов и длиной первого аргумента 16 символов). При этом худший показатель модифицированной функции всего -2,6% для третьего случая (15000 расчётов; длина первого аргумента – 16 символов, а второго – 3). Не такая уж большая потеря, особенно если учесть, что для первого случая (15000 расчётов; длина первого аргумента – 16 символов, а второго – 1) мы смогли добиться прироста производительности более чем в 30%.

Для чего я так подробно выполнил, а теперь описываю этот тест? Дело в том, что здесь мы проводили специальную оптимизацию кода, основной проблемой которой является то, что она даёт не только рост производительности для отдельных случаев работы кода, но и (зачастую) снижение производительности для некоторых других.

Такого рода оптимизация хороша тогда, когда мы знаем или можем предположить, как и для чего будет в основном использо-



» Рис. 1. Результаты тестирования производительности оптимизированной функции SXOR.

ваться наша функция. Ориентироваться нужно, конечно, на наиболее вероятные на практике случаи.

Если же достоверные данные об области применения отсутствуют, то руководствоваться необходимо данными синтетических тестов, предполагающих, что все случаи равновероятны. Однако при таком предположении всегда стоит учитывать и результаты наихудших тестов. Так, если мы видим незначительный прирост в синтетическом тесте и сильное падение в одном из специальных, то, вероятно, от такой оптимизации лучше воздержаться, так как никто не может гарантировать вам, что наихудший случай не окажется самым вероятным при практическом использовании, о котором ничего не известно заранее. Более того: я готов спорить на пиво, что именно так оно и будет.

К счастью, таких сложностей лишена общая оптимизация, при которой, как правило, растёт производительность всей функции при любом её использовании. Перейдём к ней.

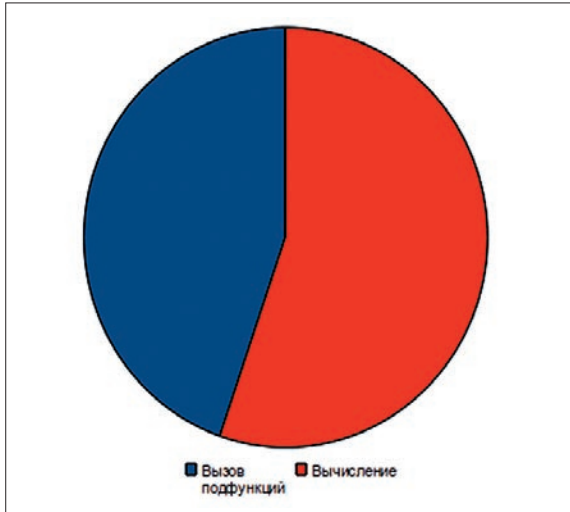
В общем случае

Для общей оптимизации обычно лучше использовать синтетические тесты, наиболее близкие к реальным условиям. Для примера здесь мы предположим, что наша функция FXOR будет использоваться преимущественно следующим образом: в половине случаев оба входящих параметра будут целыми положительными числами, а в половине – строками. Причём для строк известно, что длина первого параметра в среднем будет равна 16 знакам, а длина второго в трети случаев будет равна длине первого (то есть тем же 16 знакам), ещё в трети – одному знаку, и ещё в трети будет переменной от 1 до 4 знаков.

Для эмуляции последней трети случаев использования FXOR со строками применим преобразование значения счётчика к строковому типу – Cstr(I). При этом не забудем добавить строку **sFuncResult = Cstr(I)** в цикл измерения времени холостого хода тестера, дабы увеличить точность измерения времени работы функции.

Код же внутри цикла измерения может получиться примерно следующим:





► Рис. 2. Распределение времени на вызов и вычисление подфункции в функции COMBINE.

```
IFuncResult = FXOR(65535,0)
IFuncResult = FXOR(65535,1)
IFuncResult = FXOR(0,1)
sFuncResult = FXOR("djkyqwertyjkr",s")
sFuncResult = FXOR("djkyqwertyjkr",CStr(I))
sFuncResult = FXOR("djkyqwertyjkr",salghkmvtrdljtra")
```

Выполним замеры для исходной, неоптимизированной функции. У меня в синтетическом тесте получилось в среднем 28859 миллисекунд.

Можно переходить к оптимизации. Поочерёдно разберём все использованные приёмы.

Первое, что необходимо сделать – это избавиться от вызова подфункций. Разбиение программы на маленькие законченные блоки в виде функций или процедур очень удобно при написании кода, но сильно сказывается на производительности, так как интерпретатор затрачивает достаточно много времени на помещение процедуры в стек вызовов, инкапсуляцию переменных и другие необходимые для вызова подпроцедуры действия. Поэтому просто перенесём содержимое наших подфункций NXOR и SXOR в места их вызова из родительской функции FXOR и внесём необходимые изменения в имена переменных. В данном случае это даст нам около 6% прироста производительности. Если вам кажется, что это немного, то вспомните функцию COMBINE из предыдущей статьи, где вызов подфункции может происходить до 30 раз за вычисление. Здесь время на вызов подфункций и непосредственное вычисление результата распределено приблизительно так, как на диаграмме на рис. 2. Не слишком ли жирный кусок пирога мы отдаём за возможность использовать подфункцию, в условиях нехватки времени?

Далее можно переходить к более тонкой настройке, которая в данном случае сводится к знанию особенностей использования функции и устройства интерпретатора Basic в *OpenOffice.org*. И это наиболее важная часть данной статьи, так как этих сведений вы не встретите в других публикациях, посвящённых оптимизации кода.

Вспомним, что в функциях *Calc* невозможно передать какой либо аргумент, опустив один из предшествующих. Следовательно, проверка передачи параметра *vFirst* является излишней, так как если он опущен, то опущен и параметр *vSecond*; а значит, код проверки передачи параметров можно изменить на следующий:

```
If IsMissing (vSecond) Then ' проверяем переданы ли параметры
fXOR = "#NOTARG!" ' Аргументов слишком мало
Exit Function
End If
```

Далее, применим знание об одной не самой приятной особенности StarBasic. Рассмотрим следующую конструкцию:

```
If условие1 AND условие2 Then
код
End If
```

В большинстве руководств по другим языкам вы прочтёте, что условие2 не будет проверяться, если условие1 ложно, так как в этом случае результат всей конструкции от результата условия2 не зависит – она всё равно останется ложной.

В StarBasic это не так: условие2 здесь будет проверено в любом случае, независимо от результата условия1. Поэтому, чтобы не терять производительности, в StarBasic вышеприведённую конструкцию следует переписать так:

```
If условие1 Then
If условие2 Then
код
End If
End If
```

Проделаем эту операцию с конструкциями **If IsNumeric (vFirst) And IsNumeric (vSecond) Then...** и **If VarType(vFirst)=8 And VarType(vSecond)=8 Then...**

То, что получилось в результате этих преобразований, представлено в Листинге 3. Такой код в условиях описанного выше теста станет производительнее ещё приблизительно на 9%.

Теперь пора сравнить производительность полученной нами функции и функции, приведённой в самом начале статьи.

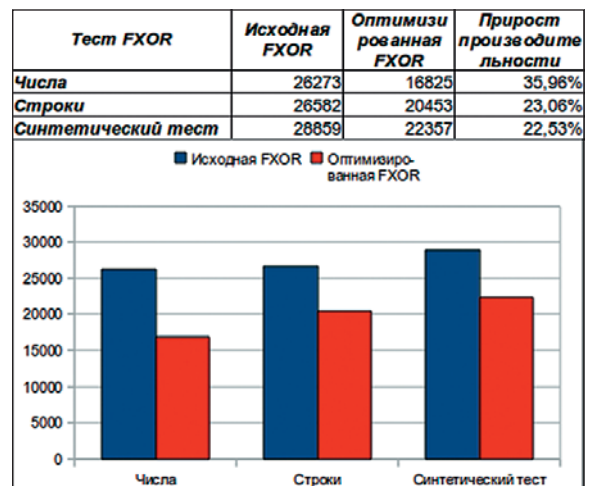
Сравнивать будем в трёх тестах – только числа, только строки и синтетический тест. Для последнего используем вышеприведённые условия. Для чисел прокомментируем строки с вызовом функции со строковыми параметрами и увеличим число проходов с 5000 до 50000. Для теста со строками прокомментируем строки с числовыми параметрами функции.

Результат тестирования на моём ноутбуке представлен на рис. 3.

Как видите, даже оптимизируя достаточно простую, и, с первого взгляда, грамотно написанную функцию, не имеющую длительных или вложенных циклов, мы добились относительно неплохих результатов, ускорив её работу в целом более чем на 20%, а в отдельных случаях – и более чем на 35%. На практике же, особенно со сложными функциями, можно добиваться гораздо более впечатляющего результата.

Общие принципы

Наконец, давайте рассмотрим некоторые общие принципы оптимизаций применительно к StarBasic. Во-первых, следите за пере-



► Рис. 3. Результат тестирования функции FXOR.

менными – старайтесь объявлять их явно. Операции с необъявленными переменными происходят медленнее. В самоконтроле хорошо помогает инструкция **Option Explicit**, добавленная в начало модуля. Контролируйте типы и избегайте их преобразования.

Используйте константы вместо переменных там, где это возможно. Константы быстрее переменных.

Используйте более экономные конструкции циклов. StarBasic поддерживает конструкцию перебора всех элементов массива **For Each**, хотя она и не документирована. Данный цикл не использует счётчик и потому быстрее цикла **For To Step**. Синтаксис конструкции таков:

```
For Each Var In Arr
    ...
Next
```

Конструкция перебирает все элементы массива **Arr**, помещая их последовательно в переменную **Var**. Используйте её всегда, когда в цикле не важен порядок перебора его элементов. Несмотря на то, что данная конструкция, как правило, перебирает элементы массива по порядку, теоретически он может быть нарушен – не стоит забывать об этом.

Минимизируйте количество операций внутри циклов. Они повторяются многократно и потому очень сильно влияют на производительность. А если вы вызываете внутри цикла другую функцию или процедуру Basic, то не удивляйтесь, что ваш код работает медленно. Вызов подпроцедур – задача очень затратная по времени. Если ваша ставка – скорость, старайтесь, по возможности, вообще отказаться от вызова стороннего кода Basic.

Отдельную тему для разговора представляют собой условия. Сами по себе они предоставляют огромный простор для оптимизации, а в отношении рассматриваемой реализации Basic – тем более. Об особенности, связанной с проверкой незначущих условий, мы уже говорили выше. А что можно сделать ещё?

Во-первых, можно попробовать отказаться от условий вообще, заменив их численным решением или выбором из массива. Простейший пример, известный по шуточному тесту, где значение переменной необходимо заменить с 1 на 2 и наоборот, даёт в численном решении пятикратный рост производительности, то есть код

```
Val = 3 - Val
в пять раз быстрее, чем
If Val = 1 Then
    Val = 2
Else
    Val = 1
End If
```

Однако это, безусловно, частный случай, и заменить условие на численное решение возможно далеко не всегда. Совсем другое дело – выбор из массива: таких решений можно найти множество, и так как в них выявляется ещё одна особенность рассматриваемой реализации Basic, разберём его подробно.

Пусть есть несложная задача: заменить числовые значения от 0 до 20 их строковыми наименованиями. Тут решение напрашивается само собой, и любая проверка условий кажется лишней:

```
Public Function MyFunction (ByVal Optional iNum As Integer) As String
    Dim aNameNum() As String
    aNameNum = Array("ноль", "один", "два", "три", "четыре", "пять", "шесть", "семь", "восемь", "девять", "десять", "одиннадцать", "двенадцать", "тринадцать", "четыренадцать", "пятнадцать", "шестнадцать", "семнадцать", "восемнадцать", "девятнадцать", "двадцать")
    MyFunction = aNameNum(iNum)
End Function
```

Однако если переписать эту функцию, используя конструкцию **Select Case** –

```
Public Function MyFunction (ByVal Optional iNum As Integer) As String
    Dim aNameNum() As String
    Select Case iNum
        Case 0
            MyFunction = "ноль"
        ...
        ...
        Case 20
            MyFunction = "двадцать"
    End Select
End Function
```

то в тесте можно обнаружить почти двукратный прирост производительности. В чём же дело? Почему проверка 4-5 условий быстрее, чем однократное извлечение из массива?

Дело в том, что мы используем интерпретатор, а это значит, что каждая загрузка массива данными – это именно загрузка: до выполнения функции **Array** он не существует нигде в виде блока данных. А загрузка массива – операция достаточно ресурсоёмкая.

Так что же, такое решение вообще не имеет права на существование? Отнюдь. И оно может даже оказаться наиболее быстрым из всех: ключевое слово здесь – кэш.

Сделаем следующее: объявим массив **aNameNum** на уровне модуля как **Global**, а саму функцию перепишем так, как в Листинге 4.

Теперь массив **aNameNum** будет загружаться данными только однажды – при первом использовании, а при всех последующих вызовах будет применяться уже инициализированный массив. Такая функция будет быстрее, чем функция, использующая сравнение, а уж свою первую инкарнацию она обогнала у меня в 2,2 раза. Результат испытаний можно увидеть на рис. 4. Естественно, такое решение подходит только там, где подобная функция может использоваться многократно.

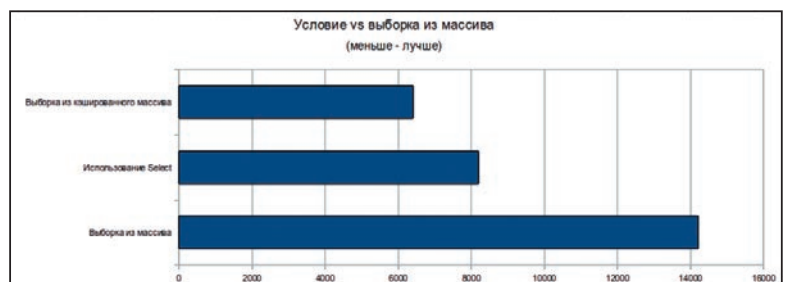
И наконец

Я обещал вернуться к вопросу о стиле программирования, предусматривающем кодирование путём исключения [Coding by exception], то есть таком, когда добавляется дополнительный код для каждого специального случая.

Безусловно, такой стиль программирования нельзя считать правильным в общем случае – код становится труден для понимания, вероятность ошибки растёт, а рефакторинг, если он потребуется, превращается в кошмар. Но считать антипаттерны недопустимыми вообще тоже было бы большой ошибкой. Мы уже видели, какой прирост по скорости может дать эксплуатация специальных случаев, а иногда это жизненно необходимо.

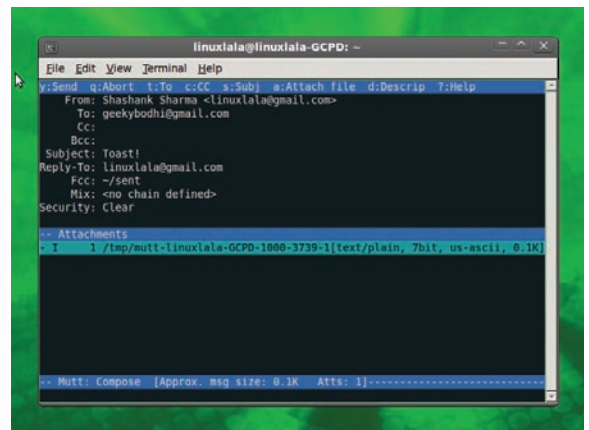
Поэтому – не пугайтесь нарушать общепринятые нормы, если знаете, что делаете: в конце концов, самый грамотный код – это тот, который работает правильно и быстро. **LXF**

➤ Рис. 4. Условие против выборки из массива.



Mutt. Почта по

Консольный почтовый клиент *Mutt* гораздо более гибок и богат настройками, чем любой другой из известных **Шашанку Шарма**. Попробуйте!



» Список необходимых горячих клавиш сверху зависит от вашего места в программе.

прочитанные сообщения в заданную папку, настроить горячие клавиши для перехода от входящих к черновикам или корзине и многое другое. Можно и настроить интерфейс в тон с обоями рабочего стола.

Установить *Mutt* можно из репозитория вашего дистрибутива. Но если вы предпочитаете стабильные версии, лучше, наверное, загрузить его прямо с сайта. Загляните в файл **INSTALL** внутри архива, для особых указаний по установочным путям и т.д.

Подготовка системы

Прежде чем пользоваться *Mutt*, нужно настроить в нем получение почты. Так как почти каждый первый и его любимый динозавр ныне пользуются IMAP, мы настроим его в *Mutt* и поработаем с Gmail.

Настройки *Mutt* хранятся в файле **.muttrc** в домашнем каталоге пользователя. Этого файла не существует, и после установки *Mutt* его нужно создать самим, а перед этим создайте каталоги **.mutt** и **.mutt/cache** в своем домашнем каталоге – откройте терминал и скомандуйте:

```
mkdir .mutt
mkdir .mutt/cache
После создания файлов запустите свой любимый текстовый редактор и заполните файл .muttrc таким содержимым:
set from = "username@gmail.com"
set realname = "ПОЛНОЕ ИМЯ"
set imap_user = "username@gmail.com"
set imap_pass = "typepasswordhere"
set folder = "imap://imap.gmail.com:993"
set spoolfile = "+INBOX"
set postponed = "+[Gmail]/Drafts"
```



Наш эксперт

Шашанку Шарма пишет о свободном ПО более четырех лет, для различных изданий, включая Linux.com. Он также является соавтором книги "Beginning Fedora" издательства Apress.

У большинства из нас масса графических программ для каждодневных нужд – например, браузеров и почтовых клиентов. Но если вы дерзнете выйти за пределы комфорта интерфейсов, управляемых мышью, утилиты командной строки предложат вам то же самое, включая электронную почту. *Mutt*, почтовый клиент для тех, кому надоели громоздкие графические программы с непомерным аппетитом к системным ресурсам, готов стать вашим лучшим другом – только позовите.

Замечали ли вы когда-либо, как в фильмах и телепрограммах показывают работающих за компьютером? Сначала демонстрируют экран монитора с необычной графикой, а потом переходят к пальцам, пляшущим по клавиатуре. Лично я пока не встречал ни графических программ со столь интенсивным употреблением горячих клавиш, ни людей, которые в таких программах вообще игнорировали бы мышшь, но как знать. *Mutt* обладает обычным интерфейсом командной строки; это означает, что все действия производятся с клавиатуры. В *Mutt* довольно легко настроить и IMAP, и POP, но на нашем уроке мы рассмотрим только IMAP.

На фоне других почтовых клиентов, *Mutt* обладает рядом преимуществ. Первое – его самокритичный девиз: «Все почтовые клиенты – морока. С нашим мороки меньше». Так как это утилита командной строки, она, очевидно, не жадна до памяти; она также богата настройками. Здесь легко перенаправить все

» Месяц назад Мы наделили будничные задачи мощью Subversion.

КОМАНДОМ



```
set trash = "imaps://imap.gmail.com/[Gmail]/Trash"
set header_cache = ~/.mutt/cache/headers
set message_cachedir = ~/.mutt/cache/bodies
set certificate_file = ~/.mutt/certificates
set smtp_url = "smtp://username@smtp.gmail.com:587/"
set smtp_pass = "ваш_пароль"
set move = no # Не спрашивать: «Переместить прочитанное в
mbox?»
set imap_keepalive = 900
```

Это простейшая и стандартная информация: об удаленной учетной записи IMAP, имени пользователя, пароле и прочем. Если в файле не будет строки

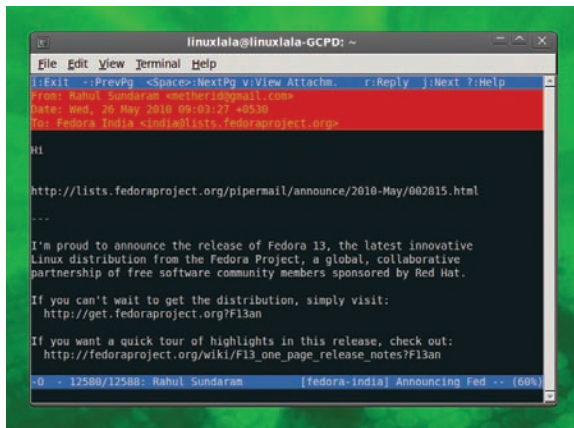
```
set header_cache = ~/.mutt/cache/headers/
```

Mutt каждый раз будет загружать заголовки всех сообщений из Входящих. Поле **set smtp_url** гарантирует, что письма, отправленные из *Mutt*, будут идентифицироваться как исходящие от законной учетной записи Gmail.

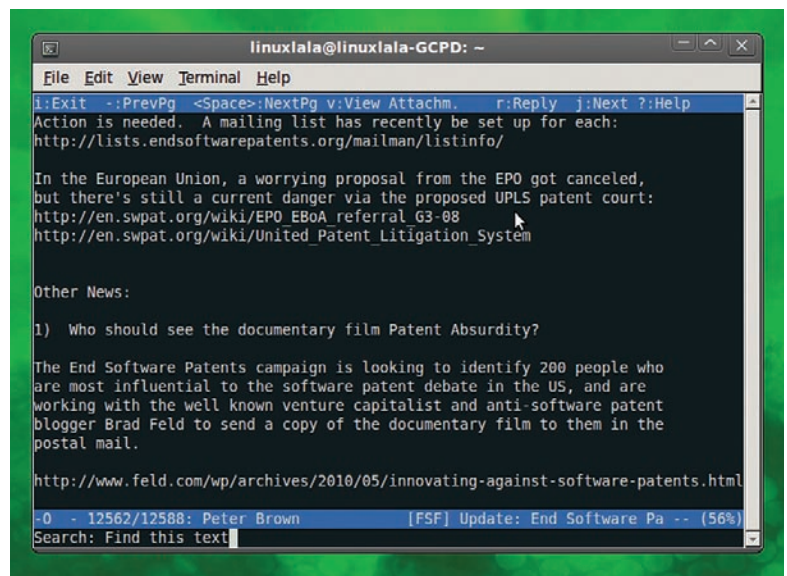
Приведенные выше настройки – почти все, что нужно для нормальной работы *Mutt*. Всего их слишком много, чтобы рассказать о них здесь, но руководства и подробная документация на сайте проекта помогут вам настроить *Mutt* согласно вашим потребностям.

Использование Mutt

Подготовив файл **.muttrc**, запустите *Mutt* из терминала, набрав **mutt** в командной строке. При первом запуске *Mutt* потребует какого-то времени для загрузки писем настроенной учетной записи электронной почты. Оповещение в левом нижнем углу терминала сообщит о статусе аутентификации и о загружаемых сообщениях.



» Пошагово с регулярными выражениями, можно обучить *Mutt* изощренному поиску.



» Вы найдете письмо во входящих или по его тексту, набрав / (прямой слэш) и введя поисковый запрос.

В зависимости от количества входящих писем, может понадобиться перелистать не одну страницу списка, чтобы добраться до самого свежего – пользуйтесь клавишами Page Up/Down. Нажатие ? покажет справку с перечислением горячих клавиш. Чтобы прочесть письмо, нажмите Enter, а потом Enter можно нажимать для прокрутки письма строка за строкой. Для перехода к следующей странице письма нажмите пробел. Нажатие I (заглавная i) вернет вас к входящим; в них можно перемещаться с помощью стрелок. Некоторые важные комбинации, скажем, такие как «Ответить» и «Написать письмо», отображаются в прямоугольнике в верхней части окна.

Поработав некоторое время с *Mutt* и почитав руководства на сайте, вы начнете понимать, какие настройки нужно добавить в файл **~/.muttrc**, чтобы программа стала для вас удобнее. Как минимум, воспользуйтесь такими опциями чтения почты:

```
# Чтение почты
set timeout=10
set mail_check=5
set sort=threads
set sort_aux=date

ignore * # Игнорировать все заголовки
unignore Subject: From: Date: To: CC:
hdr_order Subject: From: Date: To: CC:
```

Эти настройки говорят *Mutt*, как часто проверять почту, а также велят игнорировать все части заголовков писем, кроме Subject, »

Скорая помощь

Метки в Gmail превращаются в каталоги на локальном компьютере благодаря IMAP. Эта и многие другие возможности сделали IMAP более популярным, чем конкурирующий протокол POP.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

From, Date, To и CC. Поле `hdr_order` определяет порядок показа информации о заголовках.

Параметры `set sort=threads` и `set sort_aux=date` в файле `~/muttrc` покажут сообщения в виде дискуссий и упорядоченные по дате. Когда сообщения собраны в дискуссии, можно удалить их скопом, выделив любое сообщение в цепочке и нажав `Ctrl+D`. Во время сессии *Mutt* можно нажать `O` и сменить критерий сортировки. Возможные варианты будут показаны в нижней части окна. Нажатие `Shift+O` инвертирует порядок сортировки. Уже веселее, правда?

При желании просмотреть только неп прочитанные сообщения, нажмите `L` и наберите `unread`. Опция «Limit To Messages» [Ограничить Сообщениями] принимает значения `unread` [непрочитанные], `read` [прочитанные], `all` [все] и некоторые другие. Покончив с неп прочитанными сообщениями, снова нажмите `L` и наберите `all`, чтобы вернуться к входящим.

Вы заметите, что *Mutt* нумерует все сообщения, и если нужно прочесть письмо 7750, просто наберите 7750 и нажмите `Enter`. Соответствующее сообщение будет выделено, и при новом нажатии `Enter` оно откроется. Так вы сэкономите время, листая страницы сообщений во Входящих.

Если вы хотите переключаться между Входящими, Избранным [Starred], Корзиной и Черновиками во время сеанса *Mutt*, добавьте в файл `~/muttrc` такие строки:

```
mailboxes +INBOX +archive +sent +drafts +spam +trash
bind editor <space> noop
macro index,pager y "<save-message>=[Gmail]/All
Mail<enter><enter>" "Archive"
macro index,pager d "<save-message>=[Gmail]/
Trash<enter><enter>" "Trash"
macro index gi "<change-folder>=INBOX<enter>" "Go to
inbox"
macro index ga "<change-folder>=[Gmail]/All Mail<enter>"
"Go to all mail"
macro index gs "<change-folder>=[Gmail]/Starred<enter>"
```

«Можно выделить цветом письма от определенных отправителей.»

```
"Go to starred messages"
macro index gd "<change-folder>=[Gmail]/Drafts<enter>"
"Go to drafts"
```

Строка `mailboxes` велит *Mutt* отслеживать указанные папки на предмет новых сообщений. Затем с помощью свойства `macro` мы задаем горячие клавиши для переключения между различ-

ными папками и для архивации или удаления писем.

Нажатие `Enter` на выбранном сообщении перенесет вас в программу постраничного просмотра, похожую на утилиту просмотра тек-

ста *less* в *nix. Для прокрутки сообщений пользуйтесь клавишами `Backspace` и `Enter`. Большинство команд, работающих в режиме списка, работают и в режиме постраничного просмотра: например, при нажатии `J` или `K` в режиме постраничного просмотра будет показано содержимое следующего или предыдущего сообщения в списке.

Если в полученном письме есть вложение, для показа меню вложения нажмите `V`. Если вложений несколько, для перемещения между ними служат клавиши `K` и `J`. В режиме постраничного просмотра нажатие `Enter` покажет вложение только в том случае, если это обычный текст. Иначе *Mutt* попытается вызвать подходящую внешнюю программу.

Для выхода из меню вложений нажмите `Q` – вы вернетесь к постраничному просмотру. Снова нажмите `Q`, и вернетесь в режим списка.

Для связи приложений с различными форматами файлов используется файл `~/mailcap`. Например, можно связать *Evince* с PDF или *GQview* с картинками. Обычно файл `~/mailcap` выглядит таким образом:

```
application/pdf; /usr/bin/gqview %s
image/jpeg; /usr/bin/gqview %s
image/gif; /usr/bin/gqview %s
image/png; /usr/bin/gqview %s
```

В этот файл можно добавить поддержку других форматов, таких как документы *OpenOffice.org* и т.д.

Слегка расцветим

Одно из преимуществ *Mutt* – возможность выделять цветом различные сообщения, например, письма с вложениями или письма от определенных отправителей, или выделять цветом заголовки письма при его чтении. Настройки цветов также хранятся в `~/muttrc`, и, конечно, для создания подходящей цветовой схемы с ним нужно повозиться.

Обычная строка, содержащая настройки цветового выделения, в `~/muttrc` выглядит так:

```
color index blue default '~f geekybodhi@gmail.com'
```

Эта запись соответствует шаблону `color объект цвет фон [шаблон]`, а опция `~f` велит *Mutt* выделять синим цветом все сообщения с адреса `geekybodhi@gmail.com`. Точно так же, строка `color index green cyan ~N` выделит цветом все новые сообщения.

Существует уйма опций задания цветового выделения, делающего чтение сообщений более удобным. Например, строка

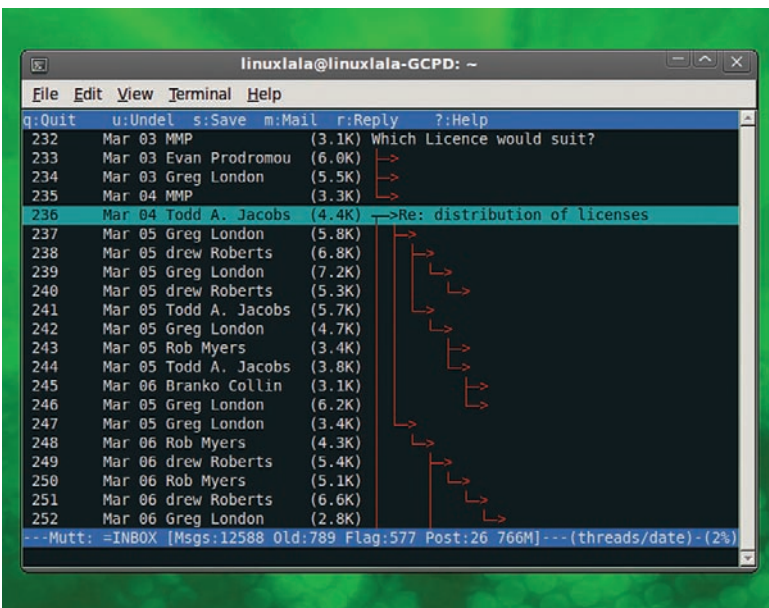
```
color attachment black red
```

выделяет цветом вложения, а следующий фрагмент выделяет заголовки письма цветом, отличным от остальной части письма:

```
color header green yellow "^Subject:"
color header yellow red "^From:"
color header yellow red "^Date:"
```

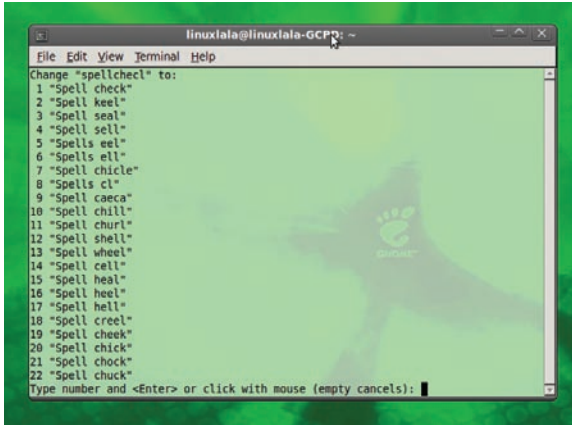


Скорая помощь
Добавьте `bind editor <space> noop` в `~/muttrc` при настройке папок, в именах которых есть пробелы.



➤ Стрелки в цепочках позволяют легко определить их в режиме списка.

➤ Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.



» Хотите верить, хотите нет, а проверка орфографии в *Vim* посрамит и *OpenOffice.org*, и большинство других программ без особого труда.

```
color header yellow red "^To:"
```

```
color header yellow red "^Cc:"
```

Мы уже задали настройки STMP в `~/muttrc`, чтобы сообщения, отправляемые из *Mutt*, идентифицировались как сообщения с удаленной учетной записи. В зависимости от дистрибутива, *Mutt* использует по умолчанию один из редакторов *Vim*, *Nano* или *Emacs*. Выбрать его по-своему можно, добавив следующую строку в файл `~/muttrc`:

```
set editor="vim"
```

Чтобы написать новое письмо или ответить на сообщение во входящих, нажмите `m` или `R` соответственно. И в том, и в другом случае откроется редактор, в котором можно набрать сообщение как обычный текстовый файл.

При нажатии `m` *Mutt* сначала спросит вас, не хотите ли вы отредактировать черновики. Нажмите `N`, чтобы создать новое сообщение. Если нажать `Y`, *Mutt* покажет все сообщения из папки черновиков, определенной в файле `~/muttrc`. При нажатии `Q` вы выйдете из черновиков и вернетесь обратно к входящим. При создании нового сообщения вас первым делом попросят ввести адрес получателя в поле **To:** в верхней части окна. Затем нужно будет ввести тему сообщения. После этого *Mutt* откроет редактор, указанный в файле `~/muttrc`. Закончив набирать сообщение, сохраните его и выйдите из *Vim* (или какой там у вас текстовый редактор).

Отправка сообщений

По умолчанию ваш текст сохраняется в каталоге `/tmp` в файле под именем `mutt-имя_хоста-xxxx-x`, где `xxxx` – цифры. Затем вы вернетесь в окно *Mutt*, где сможете проверить тему и адрес получателя, добавить адресата копии (CC), а под заголовком Attachment увидите только что сохраненное сообщение с буквой `l` слева – она означает, что вложение отправляется вместе с телом сообщения [in-line]. Таким образом, *Mutt* вкладывает сохраненный текстовый файл, содержащий ваше сообщение, в тело письма.

Если вы хотите добавить к письму еще один файл, нажмите `A`. После этого вы сможете набрать путь до файла или нажать `?` для выбора из списка. Для смены каталога нажмите `C`. После выбора файла вы снова попадете в окно просмотра. Теперь в Attachment указаны два файла. Слева от имени второго файла вместо `l` значится `A`, что означает внешнее вложение.

Чтобы отправить сообщение, нажмите `Y`. Отправив письмо, *Mutt* удаляет сохраненный текстовый файл из каталога `/tmp`. Письмо можно отправить и из терминала, командой

Проверка орфографии в Vim

Я пользовался *Vim* много лет, и больше всего мне нравится в нем встроенная проверка орфографии, введенная в *Vim 7*. *Vim* подсвечивает все ошибки и позволяет очень удобно исправить неверно набранное слово заменой из списка. Также можно добавить слово в словарь.

Все настройки *Vim* хранятся в файле `~/vimrc`. Если этого файла нет, создайте его в любимом текстовом редакторе и добавьте в него такие строки:

```
set spell
set spell spellang=en_gb
set spellfile=~/.vim/spellfile.add
highlight clear SpellBad
highlight SpellBad term=standout
ctermfg=1
highlight SpellBad term=underline
cterm=underline
highlight clear SpellCap
highlight SpellCap term=underline
cterm=underline
highlight clear SpellRare
highlight SpellRare term=underline
cterm=underline
```

```
highlight clear SpellLocal
highlight SpellLocal term=underline
cterm=underline
map <F6> <Esc>:setlocal spell
spelllang=en_gb<CR>
map <F7> <Esc>:setlocal nospell<CR>
```

Строки `map <F6>` и `map <F7>` позволяют включить и выключить проверку орфографии. После создания файла `~/vimrc` вы заметите, что все ошибки подсвечиваются и делятся на три категории – **SpellBad** (не найдено), **SpellCap** (нужно писать заглавными буквами), **SpellRare** и **SpellLocal** (сюда входит разность `Z` и `S` в американском и британском английском).

Когда проверка орфографии включена, слова с ошибками подсвечиваются. Набрав `]` `s` и `[s` соответственно, можно перейти к предыдущему и следующему слову с ошибкой. Если курсор находится в слове с ошибкой, `z=` покажет варианты подсказки. Если третье слово в списке – то, что нужно, нажмите `3` и `Enter`, и слово будет заменено. Если набрать `zg`, слово добавится в словарь, а `zug` отменит это действие.

```
mutt -s "Mail from Mutt" friendlyperson@address.com < /
home/linuxlala/msg.txt
```

Здесь содержимое файла `msg.txt` образует тело письма. Добавить к письму вложение можно и с помощью атрибута `-a`, таким образом:

```
mutt -s "Mail from Mutt, picture attached" -a /home/
linuxlala/Pictures/hackergotchi.gif friendlyperson@address.
com < /home/linuxlala/msg.txt
```

Возможность пометки сообщений позволяет выделить несколько писем и выполнить над ними какое-либо действие. Так, если вы хотите удалить пять сообщений сразу, нужно отметить их и удалить одним нажатием клавиши.

Чтобы отметить сообщение, перейдите к нему и нажмите `t`. Выделяйте сообщения до тех пор, пока не выделите все. Теперь можно выполнить действие. Для этого наберите `;` (точку с запятой) и любую команду, применимую к одному сообщению. Например, чтобы удалить одно сообщение, вы выбираете его и нажимаете `D`. А если помечены пять сообщений, можно набрать `:D` и сразу удалить все пять.

Алиасы и адресная книга

Чтобы помочь вам быстро вводить адреса, в *Mutt* есть прекрасная находка – псевдонимы (алиасы). Ее преимущество в том, что вам не нужно запоминать все адреса целиком. Простая строка

```
alias bodhi geekybodhi@gmail.com
```

(Mayank Sharma)

в `~/muttrc` сделает `bodhi` алиасом этого адреса. При создании нового письма можно просто набрать нужный алиас в поле **To:**, и *Mutt* автоматически заменит его адресом электронной почты из файла. Также можно создать алиас группы, перечислив адреса через запятую таким образом:

```
alias family member1@someemail.com,member2@
someemail.com
```

и т.д. Наконец, советую взглянуть на утилиту *Abook* на <http://abook.sourceforge.net>. Она представляет собой текстовую адресную книгу, которая работает с *Mutt*. **LEXF**

Скорая помощь

В режиме чтения письма клавиши стрелок покажут следующее/предыдущее сообщение вместо прокрутки.

» **Через месяц** Эффективно рассортируем свою почту с *Procmail*.

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru

В этом месяце мы ответим на вопросы про:

- 1 Пропавший Grub
- 2 SSH
- 3 Клон жесткого диска
- 4 Установка сервера файлов или почты
- 5 Переименование пакетом
- 6 Wine
- 7 Проблемы с модемами
- 8 Evolution
- 9 Выбор Linux-сервера
- 10 Dropbox/восстановление KDE

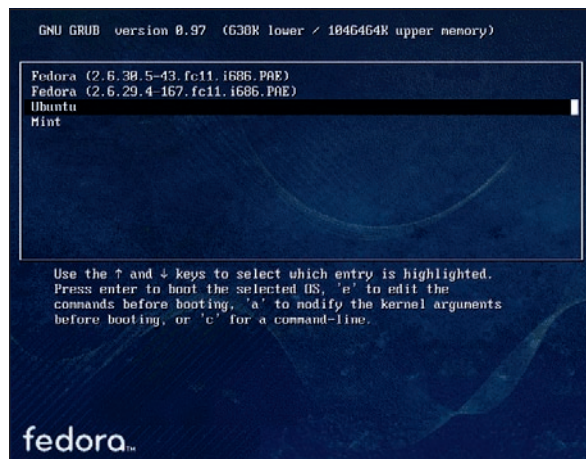
1 Проблемные разделы II

В Я новичок в мире Linux и новый подписчик LXF. Пытаюсь создать систему с двойной загрузкой на своем втором компьютере (старый Athlon с 512 МБ ОЗУ), прежде чем проделать это со своей основной рабочей машиной. Несколько попыток воспользоваться дисками от LXF и из других источников ни к чему не привели. Потом я прочел письмо в LXF131 и подумал, что наконец-то мои проблемы решены! Увы, не тут-то было. С помощью *GParted* я удалил все старые разделы Ubuntu и разделы подкачки, и даже пытался (неоднократно!) расширить мой раздел Windows на весь диск до установки Ubuntu заново. При загрузке (с установленным Ubuntu или без него) я получаю сообщение, что загрузочная

запись IDE-0 в порядке, *Grub* загружается, а потом вижу сообщение "Error: no such partition". Затем следует "grub rescue>", и машина вообще перестает реагировать на все, кроме Live CD. Может, я случайно удалил раздел подкачки Windows? Если да, то как мне его вернуть, не переустанавливая Windows? У меня есть искушение отвести этот компьютер только под Linux, но прежде чем браться за настройку нового компьютера, я хотел бы убедиться, что я действительно могу успешно создать систему с двойной загрузкой

Грэм Байерс [Graham Byers]

Удалив свои разделы Ubuntu, Вы заодно удалили файлы настройки и командные файлы, используемые загрузчиком *Grub*, но сам *Grub* остался в главной загрузочной записи (MBR). Поэтому при загрузке *Grub* тщетно пытается найти нужные ему файлы, после чего отображает минимальную командную строку восстановительного режима *Grub*. Если Вы за-



Из-за удаления разделов Linux *Grub* может не видеть файлов, нужных для нормальной работы.

тем установили другой дистрибутив, Вы можете изменить настройки *Grub* так, чтобы он использовал правильные файлы. Альтернативный вариант – еще раз удалить Ubuntu, а затем восстановить загрузчик Windows. Это позволит Вам запускать Windows, после чего Вы можете «с нуля» установить Ubuntu или любой другой дистрибутив Linux.

Чтобы восстановить загрузчик Ubuntu 9.10 или 10.04, загрузитесь с соответствующего Live CD, запустите терминал и выведите список разделов, дав следующие команды:

```
sudo -i
fdisk -l
```

Найдите раздел, помеченный как раздел Linux – назовем его `/dev/sdaX` – после чего Вы сможете примонтировать его и переустановить *Grub*, скомандовав

```
mkdir -p /media/sdaX
mount /dev/sdaX /media/sdaX
grub-install --root-directory=/media/sdaX /dev/sda
```

Ubuntu 9.10 и 10.04 используют *Grub* 2. Более ранние версии, как и большинство других дистрибутивов, используют более старую, наследуемую версию *Grub*, для которой эти команды немного отличаются:

```
sudo grub
find /boot/grub/stage1
```

Эти команды открывают оболочку *Grub* и находят местоположение файлов *Grub*. Система обозначений будет отличаться, но `/dev/sda5` соответствует (**hd0,4**), разделу первого жесткого диска (*Grub* отсчитывает номера разделов, начиная с нуля). Теперь в оболочке *Grub* можно дать следующие команды, заменив X и Y номерами, возвращенными предыдущими командами:

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 справиться со всеми вашими проблемами по части web и баз данных.



Евгений Крестников

10 лет работал системным администратором. Он поможет вам разобраться с нетривиальными вопросами настройки Linux, а также со встраиваемыми системами.



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 модернизации www.unixforum.org.

Куда посылать вопросы

Пишите нам по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxformat.ru

```
root (hdX, Y)
setup (hdX)
quit
```

Перезагрузитесь, и все должно заработать. Те, кто использует не Ubuntu или какой-либо другой дистрибутив, где применяется команда `sudo`, должны использовать вместо `sudo -i` команду `su`.

Чтобы восстановить загрузчик Windows XP, загрузитесь с восстановительного диска Windows, выберите опцию «Repair or recover» и дайте следующие команды:

```
fixboot
fixmbr
```

Для Vista и Windows 7 эти команды должны выглядеть так:

```
bootrec.exe /fixboot
bootrec.exe /fixmbr
```

Приведя загрузку в порядок, Вы снова сможете пользоваться вашими ОС или продолжить конфигурирование, установив новые. Если Вам нужна информация о наследуемом *Grub* (*Grub Legacy*) и *Grub 2*, то на **LXF DVD** Вы найдете необходимые описания и руководства в папке **/Magazine/PDFs**. **НБ**

2 Защита удаленных файлов

В Я планирую установить файл-сервер, доступ к которому клиенты Linux и Windows будут получать через защищенный туннель SSH. Хотелось бы видеть его как съемный накопитель на рабочем столе, подобно тому, как это происходит в SSHFS. Что лучше всего подходит для этого — iSCSI, AOE или USB Over IP? *Samba* не приветствуется.

Кроме того, я читала, что при разрыве сетевого соединения эффект будет тот же, что и при отключении шнура питания от жесткого диска во время операции чтения. Это правда?

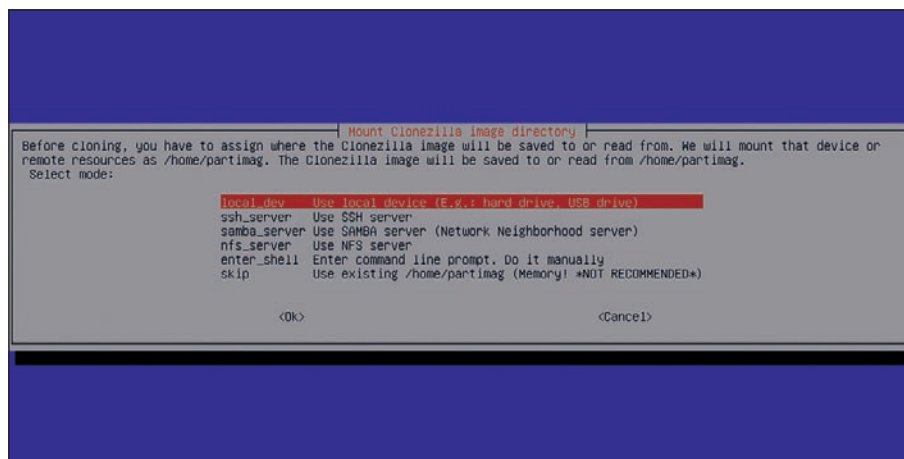
Serepe, с форумов

Технологии, о которых Вы упомянули, соединяют накопители через сеть таким образом, что они выглядят как блочные устройства. Они не для файлового сервиса, хотя Вы можете предоставлять эти устройства в общий доступ через сетевую файловую систему. Поскольку между машинами установлен SSH, почему бы не использовать его и для передачи файлов? Кроме того, Вы упомянули SSHFS, файловую систему Fuse, которая тоже хорошо работает, особенно при достаточно быстром соединении. Как вариант, можно использовать SFTP, так как сервер OpenSSH предоставляет его по умолчанию, как и SSH.

SSHFS будет лучшим выбором для монтирования каталога с сервера на базе системы Linux, потому что SSHFS работает как с SFTP, так и с SSH. Не пользуйтесь `/etc/fstab` для монтирования каталогов: монтирование от имени root не поощряется. Вместо этого выполняйте монтирование от имени обычного пользователя с помощью следующей команды:

```
sshfs пользователь@хост:каталог /точка/монтирования
```

Имя пользователя не является обязательным (по умолчанию будет использоваться то же имя поль-



Clonezilla выполняет резервное копирование, восстановление и клонирование жестких дисков или отдельных разделов.

зователя, что и ваше локальное). Имя каталога тоже не обязательно (по умолчанию будет использоваться домашний каталог пользователя). Если имя каталога начинается с /, то используется абсолютный путь; в противном случае SSHFS будет искать каталог по относительному пути в пределах домашнего каталога пользователя. Чтобы отмонтировать каталог, пользуйтесь следующей командой:

```
fusemount -u /mount/point
```

Эти команды можно запускать автоматически при запуске системы, пользуясь возможностями автоматического запуска скриптов и команд. В KDE перейдите на вкладку Advanced [Дополнительно] в окне System Settings [Параметры системы] и воспользуйтесь опцией Autostart [Автостарт], чтобы задать команды, выполняемые при запуске и остановке системы. Пользователи Gnome могут сделать то же самое через Preferences > Startup Applications [Параметры > Приложения автозапуска].

В Windows есть ряд программ, умеющих монтировать каталог SFTP так, чтобы он выглядел как жесткий диск с назначенным буквенным обозначением. Популярностью пользуется программа *Expandrive* (<http://www.expandrive.com/windows>) — но она не свободна (ни в каком из смыслов, вкладываемых в это слово). *Dokan* (<http://dokan-dev.net/en>) — реализация Fuse для Windows, включающая эквивалент SSHFS. Некоторые пользователи сообщают о проблемах с этой программой, но она бесплатна и стоит того, чтобы ее опробовать. Еще один вариант защищенного общего доступа к файлам — использование *Samba* поверх туннеля SSH, но это уже перебор, за исключением тех случаев, когда Вам действительно требуется что-то из предоставляемого *Samba*.

Насчет разрыва сетевого соединения — то, что Вы прочли, не относится к файл-серверу. Когда Вы отключаете силовой кабель, все содержимое буферов жесткого диска теряется, и файловая система остается в неустойчивом состоянии. Разрыв сетевого соединения такого влияния на диск не оказывает. При этом останавливается только передача данных из буферов (или в буфера) файл-сервера, но на сам диск это не влияет,

за исключением того, что Вы получите усеченный (но непротиворечивый) файл, запись которого велась в момент разрыва соединения. **НБ**

3 Клон жесткого диска

В Можете ли вы подсказать мне, как клонировать содержимое жесткого диска ноутбука, работающего под управлением Windows XP, или хотя бы содержимое одного из разделов этого диска, с помощью Ubuntu? Я подозреваю, что в Ubuntu за это отвечает та же система, что и за работу с жестким диском. Я сейчас в процессе перехода от XP к Ubuntu, изучаю Ubuntu, и тут возникла эта проблема. Я знаю о существовании *Acronis* и *Ghost*, но подумал, что это возможно и средствами Ubuntu.

Майк Х [Mike H]

Клонирование жесткого диска из-под работающей операционной системы — дело рискованное: природа ОС подразумевает, что в процессе клонирования файлы, расположенные на жестком диске, могут быть изменены. Поэтому самый быстрый и безопасный путь к клонированию диска — применение специального дистрибутива Live CD, содержащего инструменты специально для этой цели. Один из таких проектов — Clonezilla (<http://clonezilla.org>). Он может быть основан либо на Ubuntu, либо на Debian — в зависимости от того, какую версию Вы загрузите, и клонирование из-под него — не то же самое, что клонирование из-под вашей установленной системы Ubuntu. Clonezilla — это набор инструментов для резервного копирования и восстановления, скомбинированный с набором вспомогательных скриптов, которые делают процессы резервного копирования и восстановления дисков и разделов простыми и удобными.

Скачайте ISO-образ и прожгите его на CD, как Вы делаете обычно. Затем загрузитесь с этого CD, и Вы увидите текстовое меню; оно проведет вас по всем шагам резервного копирования и восстановления образов дисков. Обычная процедура — сохранить образ диска в файл, копируемый на другой диск, после чего воспользоваться Clonezilla для восстановления этого образа на новый диск. **»**

Одновременно подключив старый и новый диски, Вы можете провести снятие образа непосредственно с диска на диск, без создания промежуточных файлов. **ПХ**

4 Гибкий файл-сервер

Мы используем для нашего файл-сервера Synology DS207+, а остальная сеть состоит из настольного компьютера с Ubuntu 9.10 и ноутбука с Windows 2000, соединенных друг с другом через маршрутизатор Netgear DGVF338. У нас малое предприятие, да еще мы используем эти компьютеры для домашних целей. Хотелось бы иметь возможность приема и отправки электронной почты с любого из компьютеров, с помощью *Evolution* (на компьютере Linux) или *Outlook* (на компьютере Windows). Мы хотели бы синхронизировать обе почтовые программы, и по отправленным, и по полученным сообщениям. Возможно ли это? Можно ли использовать DS207+ как разновидность сервера? Наш провайдер — Virgin Media, и выделенного IP-адреса у нас нет.

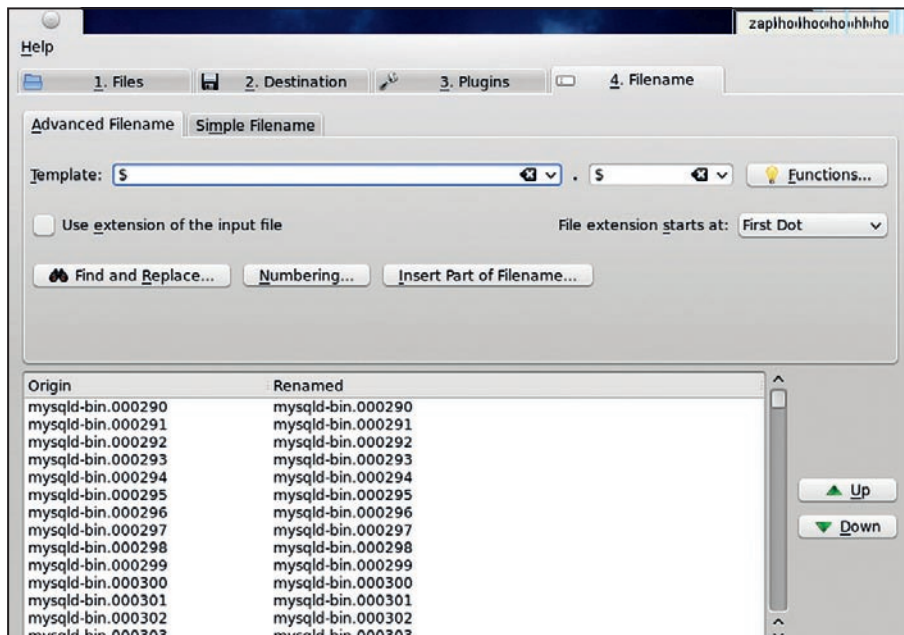
Алан Эдвардс [Alan Edwards]

В принципе это возможно. Нужно настроить DS207+ в качестве почтового сервера. Поскольку у Вас нет статического IP-адреса, Вы не можете получать почту прямо на сервер; Вам потребуется установить *Fetchmail* и настроить его так, чтобы производить опрос почтового сервера вашего провайдера на предмет поступающей почты и скачивать все новые сообщения. Как только новые сообщения попадут на сервер, Вы можете воспользоваться сервером IMAP — *Dovecot* или *Cyrus-imapd* — чтобы сделать почту доступной для компьютеров из вашей сети. При использовании протокола IMAP вся почта хранится на сервере (клиенты хранят только кэшированные копии), и статус каждого сообщения (например, прочитано оно или нет) тоже хранится на сервере. А значит, прочитав почту с одного компьютера, а затем переключившись на второй, Вы увидите то же самое, как если бы продолжали работать на первом компьютере. Это действует не только для разных компьютеров, но и для системы с двойной загрузкой — одна и та же почта будет видна в обеих операционных системах.

Почтовые клиенты будут сохранять копии всех исходящих сообщений — обычно для этого используется папка Sent на сервере IMAP; и читать ваши отправленные сообщения можно будет с обоих компьютеров. Если Вы хотите, чтобы устройство DS207+ обрабатывало исходящую почту, установите и сконфигурируйте *Postfix*. У нас здесь нет места на детальное описание всех трех программ, но они обсуждались в **LXF115**. Кроме того, на форумах *Synology* есть ветка, где установка этих программ на DS207+ обсуждается подробно: <http://forum.synology.com/enu/viewtopic.php?f=32&t=10167>. **НБ**

5 Модем в Ubuntu пакостит

В связи с последним письмом о проблемах с 3G в Ubuntu, создается впечатление, что заставить устройства 3G Huawei работать



► *KRename* может переименовать файлы пакетом, даже если они сидят на удаленном сервере.

под Linux — это относительно обычная и простая задача. А вот лично мне ни разу не удалось заставить работать под Linux мое устройство Huawei E1550 от O2: система всегда видит его только как накопитель. Я пользуюсь Ubuntu 9.10 на Lenovo S10e. Прилагаю содержимое соответствующих команд и системных журналов.

Шон Миллер [Sean Millar]

Похоже, что Huawei нарочно осложняет жизнь пользователям Linux. Под Linux для большинства их модемов достаточно извлечь [eject] устройство, самоидентифицирующееся как CD-ROM, и модемы появятся. Теперь они продвинулись на этом пути еще на шаг вперед: сейчас устройство выглядит как съемный жесткий диск. В Windows, после установки драйверов для этого устройства, при следующем его подключении драйвер отправляет команду на переключение устройства из режима накопителя в режим модема, и идентификаторы USB меняются (это бывает полезно потом, когда Вы при помощи *lsusb* сможете определить, переключили ли устройство).

Чтобы проделать это под управлением Linux, есть специальная программа: *USB_ModeSwitch* (<http://draisberghof.de/usbmodeswitch>). Она может быть уже включена в состав репозитория вашего дистрибутива; в противном случае можно установить ее из исходного кода. Программа отправляет команду на переключение устройства из-под Linux и может вызываться из правила *udev*, чтобы переключение сделалось автоматическим. Но Вам по-прежнему необходимо знать код, который нужно отправлять и который помещается в */etc/usb_modeswitch.conf*. В этот файл нужно поместить следующие строки:

```
DefaultVendor = 0x12d1
DefaultProduct = 0x1446
MessageEndpoint = 0x01
```

```
MessageContent = "555342430000000000000000
0000000110600000000000000000000000000000"
```

Теперь установите правило *udev*, создав файл */etc/udev/rules.d/10-huawei1550.rules*, содержащий следующее:

```
SUBSYSTEMS=="usb",
SYSFS(idProduct)=="1446",
SYSFS(idVendor)=="12d1", RUN+="/usr/sbin/
usb_modeswitch"
```

При подключении модема должно сработать правило *udev* — оно изменит тип устройства на модем, а *lsusb* покажет другой идентификатор продукта. В противном случае попробуйте скопировать вручную

```
sudo usb_modeswitch
```

и посмотреть, не поможет ли это. Если да, еще раз проверьте свое правило *udev*. Если нет, Вам может потребоваться отправить другой код, который Вам придется выяснить с Windows.

Установите *SniffUSB* (<http://www.pcausa.com/Utilities/UsbSnoop/SniffUSB-x86-2.0.0006.zip>), затем установите драйверы для модема. Запустите *SniffUSB* и подключите модем, выберите опцию «List Devices Not Present», потом свой модем, и нажмите Install. Отключите модем и подключите его снова. Подождите несколько секунд, выйдите из *SniffUSB*, скопируйте созданный программой файл журнала на флэш-накопитель и снова перезагрузитесь в Linux. Найдите в файле строки, начинающиеся с 'TransferBufferMDL'. Такая строка должна найтись сразу же после раздела, начинающегося с

```
PipeHandle...
TransferFlags...
```

Под ней будут находиться две строки чисел, примерно такого вида:

```
00000000: 55 53 42 43 90 4e d6 8a 24 00 00 00 80
00 08 ff
00000010: 02 44 45 56 43 48 47 00 00 00 00 00
00 00
```

Удалите части строк перед символами двоеточия и все пробелы и объедините оставшиеся строки в одну. Ее надо занести в параметр 'MessageContent' в `/etc/usb_modeswitch.conf`. Строка `PipeHandle` (см. выше) должна содержать значение конечной точки – это значение `MessageEndpoint` для `/etc/usb_modeswitch.conf`.

Поддержка на уровне ядра для этих устройств улучшается, и необходимость запуска `usb_modeswitch` скоро отпадет (хотя это и не повредит, потому что идентификаторы USB ID уже изменились) – до тех пор, пока производители не выпустят новые модемы с еще более безумными настройками. **НБ**

6 Медленный Wine

В Не прольете ли вы свет на вопрос, почему *Wine* в моей системе стал так медленно запускаться? Чтобы сделать хоть что-то, может уйти больше минуты. После запуска *Wine* обычно работает уже хорошо. Для музыки я использую программу *Spotify*, и с ней творится примерно то же. В *System Monitor* процесс отображается как 'pipe_wait', как и все процессы, имеющие отношение к *Wine*.

Нельзя ли наскоро залатать эту проблему, до подхода нового релиза Linux Mint с ускоренной реакцией?

Loz

О Сигнал `pipe_wait` означает, что программа ожидает данных, либо для отправки, либо для получения. Поскольку Ваша проблема имеет отношение к *Spotify*, скорее всего, эта программа ожидает поступления данных с сервера через сеть. Некоторые версии *Wine* более чувствительны к этой проблеме, чем другие, и стандартный совет – обновиться до новейшей версии (сейчас это версия 1.1.43). Некоторые ди-

стрибутивы включают в репозитории только старые серии 1.0 (похоже, в Mint это именно так), но на самом деле Вам необходимы версии из ряда 1.1. В архивах Ubuntu (PPA) для *Wine 1.1* есть необходимый пакет, который можно добавить с помощью следующих команд:

```
sudo add-apt-repository ppa:ubuntu-wine/
ppa
sudo apt-get update
```

Этот PPA содержит версию 1.1.42, хотя к тому моменту, когда Вы будете читать наш ответ, может стать доступной уже новая версия. Если Вам требуется версия еще новее, придется выполнять сборку из исходного кода. Впрочем, сперва попробуйте версию из PPA – 1.1.42 всяко новее 1.0. **НБ**

7 Массовое переименование

В Я только что поменял имя одного из моих узлов *Munin* в файле настройки на нечто более осмысленное. Там хранятся данные истории, и имена файлов содержат строки `localhost-etc*`, например:

```
localhost-sendmail_mailstats-discarded-d.rrd
Теперь я хочу переименовать все вхождения localhostX в serverX. Нет ли варианта наподобие 'for i in', чтобы сделать это?
```

LeeNukes, с форумов

О Как нередко в Linux, решений тут несколько. Можно сделать это с помощью цикла `for`, сначала внося изменения в каждое имя с помощью `sed`, а затем передавая их команде `mv`. Но это – сложный подход, чреватый ошибками. Если Вы работаете с KDE, можно воспользоваться *KRename*, GUI-утилитой, созданной специально для пакетного переименования файлов. Выделите файлы, которые требуется переименовать, задайте правила переименования и запустите процесс. Утилита выполняет свою задачу, даже если файлы на-

ходятся на сервере, потому что KDE использует протокол Fish для доступа к удаленным каталогам через SSH, и Вам также требуется доступ к серверу через SSH, чтобы использовать удаленные альтернативы командной строке.

Но простейший способ – применить команду `rename`; она уже установлена на сервере, так как является частью пакета `util-linux`. Эта команда предоставляет аналоги `dmesg` и многое другое. Команда

```
rename localhostX serverX *.rrd
```

выполнит переименование на одном сервере. Если Вам так уж хочется использовать цикл `for`, обработайте в нем множество имен хостов:

```
for i in X Y Z; do
  rename localhost$i server$i *.rrd
done
```

Существует альтернативная команда `rename`, доступная по адресу <http://rename.berlios.de>, с более продвинутыми функциями переименования на основании регулярных выражений. С ее помощью вашу задачу можно решить так:

```
rename -s/^localhost/server/ localhost*.rrd
```

Кроме расширенных возможностей по переименованию, эта команда предоставляет опцию `--test`, позволяющую заранее проверить, что получится в результате Ваших действий и убедиться, что вы не грохнете каталоги с жизненно важными данными. **MC**

8 Ubuntu TalkTalk не так

В Известны ли проблемы с подключением Ubuntu 10.1 к сети TalkTalk через *Evolution*? Я не получаю почтовые сообщения, хотя могу их отправлять. Все было в порядке с версиями Ubuntu от 7 до 9, а с версией 10.1 начались проблемы. Я могу получать электронную почту на ноутбуке с Windows XP.

Дес Вуд [Des Wood]



Часто задаваемые вопросы

Fish

» Это насчет нудных постов на форумах *Linux Format*?

Нет, это серьезная и полезная тема, которой соответствует слегка неочевидный акроним: Files transferred over SHell (файлы, переданные через протокол SHell).

» Звучит не особо волнующе. И что эта штука делает?

Просту предоставляет вам доступ к файлам на удаленном компьютере только через SSH (или RSH). Удаленный компьютер не нуждается в специализированном ПО – требуются только SSH и стандартные команды оболочки. На вашем локальном компьютере требуется только ПО с возможностями Fish для доступа к удаленным файлам.

» А где найти такое ПО?

Вполне возможно, что оно у вас уже есть. Например, с Fish работает *Midnight*

Commander. Войдите в него и наберите команду

```
cd /#sh:user@hostname/some/path
```

и увидите содержимое каталога `/some/path` на удаленном хосте. Если вы опустите строку `user@`, команда возьмет текущее имя пользователя.

» Значит ли это, что кто попал может читать файлы с моего компьютера при помощи Fish?

Нет, потому что Fish использует SSH. Для регистрации в системе нужен пароль, если только вы не предпочли аутентификацию по ключам. В обоих случаях, сделать это может лишь тот, кому вы разрешили.

» Хорошо, но *Midnight Commander* старомоден, не так ли? Как насчет более современного GUI?

У вас KDE? Если да, откройте *Konqueror* (или мини-консоль, клавишами Alt+F2) и введите

```
fish://user@hostname/some/path
```

Благодаря магии `gio_slaves`, эта команда загрузит в *Konqueror* удаленный каталог. В сочетании с возможностями *Konqueror* по разбиению окон, получается удобный способ сравнения локальных каталогов и их удаленных копий, а также другие варианты работы с ними.

» Значит, так я могу управлять файлами в MC или *Konqueror*. И все?

Вовсе нет. Fish не ограничивается только управлением файлами. Любая программа KDE, имеющая доступ к файлам, может получать к ним доступ и через Fish. Например, вот как можно редактировать файл на вашем Web-сервере, который располагается где-нибудь на другом краю света:

```
kate fish://me@my.server/iffypagename.html
```

Файл загрузится в *Kate*, и как только вы выберете опцию Save [Сохранить], модифицированный вариант будет отослан обратно на сервер. Разумеется, это касается редактирования не только web-

страниц; точно так же удобно править, например, файлы настройки. Fish можно даже использовать с графической утилитой для сравнения файлов на удаленном сервере.

» И что, из Fish можно управлять файлами на удаленных серверах, пользуясь учетными записями типа предоставляемых *Rsync.net* или *Strongspace*?

Увы, нет. Fish работает только тогда, когда вы регистрируетесь в системе и выполняете базовый набор команд оболочки. Названные сервисы не предоставляют полную регистрацию, дающую доступ к командным оболочкам, поэтому Fish работать не будет. Но есть альтернатива: *SSHFS* (<http://fuse.sourceforge.net/sshfs.html>) – файловая система *Fuse*, которая позволяет монтировать файловые системы, предназначенные только для копирования файлов (`scp` или `rsync`); правда, потребуется монтировать удаленную файловую систему, даже если нужно скопировать всего один файл.

○ А не пользуетесь ли Вы нестабильным Ubuntu 10.10? В конце апреля была выпущена версия 10.04. Evolution для получения почты использует стандартные почтовые протоколы POP3 и IMAP. Хотя Вы и не указали, каким из них пользуетесь, но оба должны работать нормально. Еще один вопрос: что происходит, когда Вы пытаетесь принять почту? Получаете ли Вы сообщения об ошибках? Чаще всего ошибки возникают из-за опечаток при вводе имени пользователя и пароля, чувствительных к регистру вводимых символов. В результате появляется диалоговое окно «Authentication Failed» [Сбой аутентификации]. Перепроверьте, правильно ли Вы ввели имя пользователя и пароль; если Вы не уверены в пароле, можете зарегистрироваться в TalkTalk Dashboard по адресу <https://supportcentre.talktalk.net> и изменить его, или обратиться по телефону в службу техподдержки клиентов.

Если пароль тут ни при чем, откройте меню справки Evolution, выберите команду Debug Logs [Отладочные журналы] для просмотра журналов и установите опцию Log Messages [Регистрировать сообщения] в Warnings and Errors [Предупреждения и ошибки]. Затем еще раз попробуйте принять почту и посмотрите, что выводится в окно журнала. Подобные сообщения бывают несколько загадочными, но ошибки выделены четко, и они могут дать ключ к решению проблемы. Если ваша проблема по-прежнему не решается, попробуйте поискать по тексту сообщения в одном из поисковиков, или же отправьте его нам по адресу answers@linuxformat.ru или на форумы LXF: <http://forum.linuxformat.ru>. **НБ**

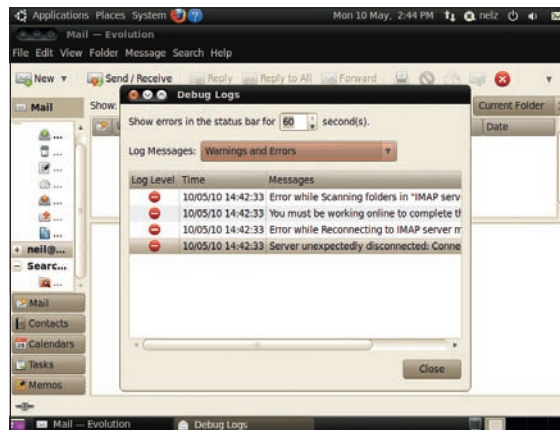
9 Планирование сервера

В Я планирую собрать сервер 17 с таким объемом RAM, какой только смогу установить, поскольку у меня пять одиночных компьютеров с Microsoft XP или Windows 7 с сопутствующими проблемами. Надеюсь, что смогу подключить к сети большинство из них (желательно все) через сервер Linux. В этом-то и моя проблема, потому что я ничего не знаю о Linux и о том, какая комбинация аппаратных средств и ПО подходит для поставленной цели.

Судя по тому, что я слышал, Linux позволяет работать ОС от Microsoft, но не наоборот; и потом, вариантов Linux очень много, и я затрудняюсь выбрать подходящий. Не подтолкнете ли вы меня в нужном направлении? Только, пожалуйста, объясняйте попроще.

Фил [Phil]

○ Прежде всего Вам нужно точно определить, что именно Вы хотите от этого компьютера. Если Вы хотите организовать файл-сервер в вашей сети из Windows-клиентов или даже главный контроллер домена, то Linux отлично справится с этой задачей при помощи программы Samba. Если Вы хотите предоставлять другие сервисы, например, почтовый или web, то и на это есть программы. Ваши сведения о том, какая ОС кому что позволяет, вероятно, относятся к системам с двойной загрузкой, когда на одном



➤ При проблемах с доступом к электронной почте первым делом загляните в файл журнала Evolution.

компьютере устанавливается две (или более) операционных систем, а потом ОС выбирается при загрузке компьютера. Возникающая здесь проблема зависит от установщиков соответствующих операционных систем. Установщики Linux обнаруживают наличие Windows и должным образом настраивают загрузочное меню, а вот Windows воображает себя единственной ОС во Вселенной и затирает загрузочную информацию имеющихся на компьютере инсталляций Linux. Впрочем, затирается только загрузочное меню, а не сама операционная система; тем не менее, желая создать мультизагрузочную систему, проще сперва установить Windows, а уж затем Linux (LXF133).

Если Вы хотите использовать этот компьютер под Linux-сервер и не намерены устанавливать на него никаких других ОС, Вам не о чем беспокоиться: потребуется только подходящий дистрибутив Linux. С задачей справится большинство дистрибутивов, но для серверного использования лучше всего Ubuntu Server Edition или CentOS. А если Вам хочется установить Linux и на настольных компьютерах, устанавливайте эту ОС после Windows и дайте инсталлятору возможность перемасштабировать раздел Windows, чтобы выделить пространство под установку и создать систему с двойной загрузкой. Здесь, как и в любом случае, Вы имеете полную свободу выбора дистрибутива, однако единообразие между клиентами и сервером упростит Вам изучение незнакомой ОС.

Linux ныне поддерживает большинство аппаратных устройств – исключением являются только новейшие модели, для которых еще не написаны драйверы уровня ядра. Обычно, чтобы выяснить, совместимо ли с Linux оборудование, которое Вы собираетесь приобрести, достаточно минут пять провести в поисковике. Основные устройства, на которые надо обращать внимание – это материнские платы, сетевые адаптеры и графические карты. Выполните поиск, указав номер модели вместе с ключевым словом 'Linux', и обнаружите, что по этому поводу пишут другие. Впрочем, обращайте внимание и на даты – например, если с момента публикации поста наподобие «материнская плата XYZ не поддерживается в Linux» прошло уже примерно полгода, то эта информация, скорее всего, уже устарела. **НБ**



Коротко про...

Exif

В вопросах, публикуемых в этой рубрике, не раз упоминался Exif; и что же это такое? Exif (сокращение от EXchangeable Image File) – стандарт на включение метаданных в файлы определенного типа, например, JPEG и TIFF. Exif используется цифровыми камерами для хранения, наряду с самим изображением, детальной информации о фотографиях: сюда входят дата и время съемки, сведения о камере (модель, серия и номер), настройки экспозиции и вспышки и (возможно) еще многое другое. Некоторые камеры сохраняют даже имя фотографа, а то и информацию GPS-навигации.

В KDE щелкните правой кнопкой мыши по файлу с фотографией, откройте окно свойств – и увидите базовую информацию Exif. Многие средства просмотра тоже отображают эту информацию, и ее можно также просмотреть из командной строки с помо-

щью программ типа Jhead (<http://www.sentex.net/~mwandel/jhead>) или Exiftool (<http://www.sno.phy.queensu.ca/~phil/exiftool>). Утилиты командной строки хороши тем, что их вывод можно обрабатывать скриптами или инструментами манипуляции текстами (вроде *grep* и *awk*), что упрощает извлечение конкретной информации. Кроме того, они позволяют изменять данные Exif.

Самое популярное применение для этих возможностей – ситуация, когда Вы уехали в отпуск и забыли перенастроить камеру на другой часовой пояс. Потом, посреди отпуска, Вы вдруг обнаруживаете, что на доброй половине ваших фотографий указано неверное время. Беде поможет Exiftool. Например, для всех .jpg-файлов в текущем каталоге время на 3 часа вперед переведет следующая команда:

```
exiftool -AllDates+=3 *.jpg
```

А на столько же часов назад переведет время во всех файлах ваших фотографий в каталоге photos/holiday, включая все его подкаталоги, команда

```
exiftool -r -AllDates-=3 photos/holiday
```

БОЛЬШОЙ ВОПРОС Как запустить Dropbox в KDE?

10 Dropbox «роняет» KDE

В Я несколько лет время от времени работаю с Linux, и большую часть своих разработок (в Ruby on Rails) делаю на машине с Kubuntu. Все шло хорошо, пока я не решил установить *Dropbox* на Kubuntu. Сначала вообще ничего не вышло, но затем появился Deb-пакет для Ubuntu, и я решил, что он нормально установится с помощью *dpkg*. Я попытался, но тут пошли жалобы на отсутствующие зависимости, в основном связанные с *Nautilus*.

Я «продрался» через эти препятствия, добывая каждую зависимость по мере появления сообщений об ошибках; вывод *dpkg* рекомендовал мне использовать опцию *-f*, что я и сделал. И зря! Установка прошла успешно, но теперь после перезагрузки моя машина выглядит, как машина Gnome. Все мои программы KDE вроде работают, но я хочу вернуть мой рабочий стол KDE. Нет ли какого-нибудь способа, кроме переустановки?

Йон Зайдель [Jon Seidel]



Клиент *Dropbox* для Linux включает модуль расширения *Nautilus*, а тот, в свою очередь, требует установки боль-

шей части компонентов Gnome. К ним относится и *GDM*, *Gnome Display Manager*, который выводит регистрационное окно и запускает рабочий стол. Если Ваша система настроена на автоматическую регистрацию, временно блокируйте эту возможность, чтобы увидеть окно регистрации в системе. Ваш первый шаг – запустить *Synaptic*, чтобы деинсталлировать *Dropbox* и большинство пакетов Gnome, установленных вместе с ним. В частности, позаботьтесь об удалении *GDM*. Чтобы перенастроить KDM, запустите команду `sudo dpkg-reconfigure kdm`

Удалив ненужные пакеты Gnome, Вы снова получите регистрационный экран KDM и рабочий стол KDE.

Так Вы вернете свою систему в исходное состояние, до установки *Dropbox*. Конечно, неплохо было бы установить клиент KDE для *Dropbox*, но, поскольку это закрытое ПО, нам приходится полагаться на то, что дает нам *Dropbox*. Однако заставить *Dropbox* работать с KDE вполне реально. Загрузите последнюю версию файла *tar.gz* с сайта <http://www.getdropbox.com/download?plat=lnx.x86> (в случае 64-разрядной версии добавьте *_64*).

Распакуйте архив в свой домашний каталог; создастся каталог *.dropbox-dist*. Теперь, чтобы запустить демон *Dropbox*, скомаундите `~/dropbox-dist/dropboxd`

Демон ругнется на невозможность обнаружения *Nautilus*, но все же будет работать, выводя инструкции по предоставлению информации о Вашей учетной записи или предлагая создать ее, если у Вас ее пока нет. Кроме того, в системном лотке появится значок *Dropbox*; щелчком правой кнопкой мыши по этому значку Вы получите доступ к имеющимся функциям. Чтобы демон запускался при загрузке KDE, выберите опции меню System Settings > Advanced > Autostart, затем команду Add Program, и наконец выберите значок для *dropboxd* оттуда, куда Вы его поместили.

Обратите внимание, что нужен весь каталог *.dropbox-dist* – нельзя просто скопировать *dropboxd* в произвольный каталог и надеяться, что он будет работать. Кроме того, есть еще командный сценарий для работы с *Dropbox*, который можно скачать отсюда: <http://forums.dropbox.com/topic.php?id=2543>. Пройдите по ссылке на dbcli.py. **НБ**



Шаг за шагом: Настроим Dropbox



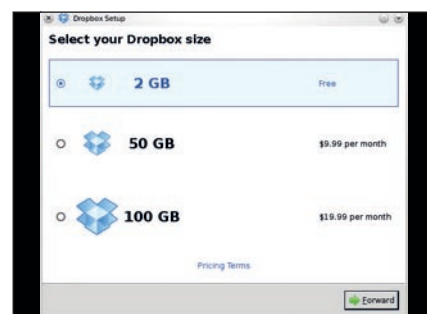
1 Настройки Dropbox

При первом запуске *dropboxd* он запросит информацию о вашей учетной записи или, при отсутствии таковой, даст вам возможность ее создать.



2 Создание учетной записи

Если у вас еще нет учетной записи, создайте ее сейчас; впоследствии вы сможете добавить к той же учетной записи и другие компьютеры.



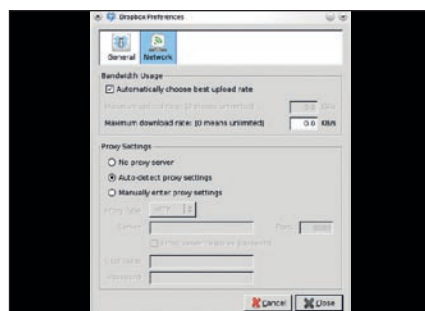
3 Бесплатно как пиво

Базовая учетная запись *Dropbox* бесплатна, как и сам продукт. Но это не Open Source.



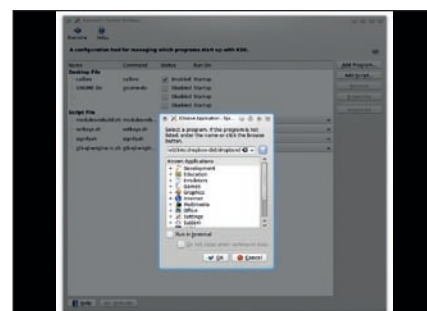
4 Работа с KDE

Хотя прямой интеграции с файловым менеджером KDE нет, пользователи KDE все же получают значок в системном лотке и управляющее меню.



5 Настройка параметров

Из системного лотка вы можете добраться до параметров *Dropbox*. Возможно, вы захотите ограничить скорость закачки, чтобы *Dropbox* не съел всю вашу полосу пропускания.



6 Автозапуск

Добавьте *dropboxd* в Autostart через меню KDE System Settings, чтобы при входе в систему демон загружался автоматически. **LXF**

LXF HotPicks



Ник Вейч

Ник компилирует HotPicks, используя текстовый редактор и головной мозг. А иногда — только текстовый редактор.

JStock » Open Cubic Player » XMLTV » Presage » KeePassX » Tubularix
» Pasang Emas » Fritzing » Di » Padre

Программа слежения за рынком ценных бумаг

JStock

Версия 1.0.5 Сайт <http://jstock.sourceforge.net>

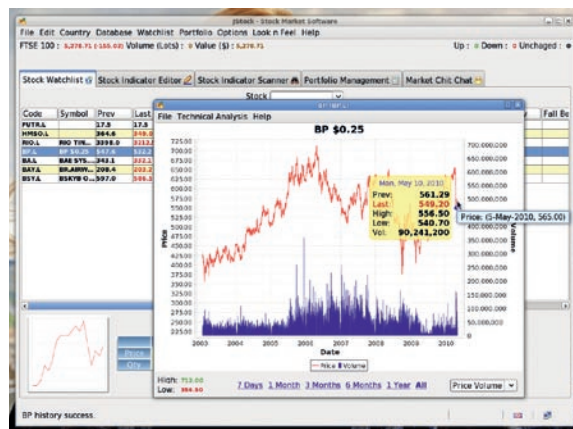
Маргарет Тэтчер продвигала общество акционеров. Гордон Гекко [Gordon Gecko, герой фильма «Уолл-стрит», — прим. пер.] продвигал красные подтяжки (однако за обедом был не так уж горяч). Даже те, кто сильно подвержен ностальгии, признают, что восьмидесятые не вызывают особо пылкой любви; зато они вывели немало людей на идею рынка ценных бумаг и торговли акциями.

Хотя, возможно, анализ за завтраком сырьевого рынка зерновых культур — это не есть хорошая игра в вашем понимании, большинству известно, что финансовый рынок выживает за счет информации. Цены на акции поднимаются и опускаются быстрее, чем рука у Майка за выпивкой, и причиной колебаний может стать всего-навсего

слух о некоем событии, которое может, и не произойдет. Отслеживание ваших инвестиций может оказаться решающим.

А для любителей таблиц это может стать еще и удовольствием. Программ для слежения за рынком ценных бумаг много, но в их толпе заметна *JStock* — свободная программа с открытым кодом на Java; она поладит с любой из основных платформ — что вскоре станет более значимым.

Магия заключается в получении данных о рынках ценных бумаг через API Yahoo!, и хотя информация и не сиюминутная, она вполне сойдет для всех, кроме, возможно, самых матерых пользователей. Вы можете не просто перечислить текущие цены на акции, но и ввести транзакции по продаже и покупке, чтобы отслеживать порт-



» Нефтяные разливы вредят бизнесу, если только вы не вложились в компанию, занимающуюся технологиями очистки.

фолио. Очень разумно, что при этом учитывается стоимость брокерских услуг, так что вы сможете увидеть истинный доход и убыток от своей рыночной активности.

Тары-бары

Пощелкайте мышью — и перед вами окажется график с хронологией событий по любым акциям, позволяющий планировать свои торговые операции в соответствии.

Симпатичное дополнение — канал чата. Он реализован через Jabber и позволяет общаться с другими пользователями *JStock* по поводу текущих тенденций, состояния рынка или того, чего у вас нет на обед. Самому-то мне не удалось найти ни одного желающего поболтать, но, возможно, всех распугал мой «злбный» ник. Возможно, вам повезет больше.

Совершенно убойная функция, по крайней мере для приложения подобного рода — хранение данных «в облаке». Работая (что очевидно) как проект Google App Engine, он хранит ваши адреса электронной почты и просматривает рынок ценных бумаг/портфолио в режиме онлайн, так что вы сможете видеть их повсюду на любом совместимом устройстве при наличии Java. Вы можете работать со своими ценными бумагами дома, в офисе (естественно, если политика вашей компании этого не запрещает) или даже на дне септического резервуара в Шропшире, если он входит в зону покрытия мобильного Интернета. Чего ждем? Покупай, покупай, покупай!

Исследуем интерфейс Jstock

Выбор страны

Перед тем, как начать загружать данные по рынку ценных бумаг, проверьте, указали ли вы свою страну.

Обзор рынка

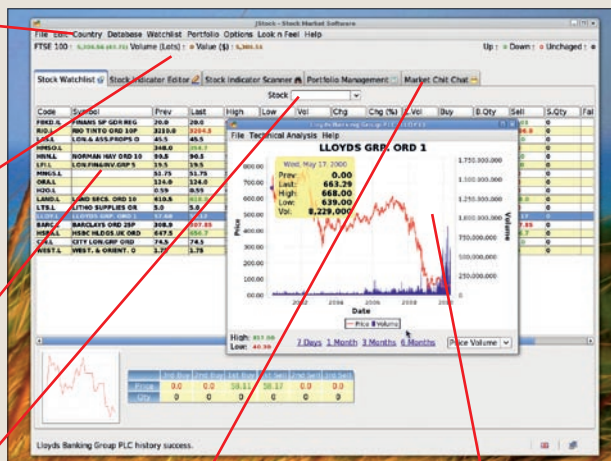
Обзор рынка появится здесь, и будет обновляться в режиме почти реального времени.

Листинги ценных бумаг

Данные для листингов берутся в Yahoo с минимальной задержкой.

Поиск

Просматривайте ценные бумаги с помощью умной функции поиска.



Чат

JStock включает клиент *Jabber* для общения с другими пользователями.

Визуализация

Создавайте графики для выбранных акций, щелкнув правой кнопкой и выбрав *History*.

Музыкальный проигрыватель

Open Cubic Player

Версия 0.1.20 Сайт <http://stian.cubic.org/project-ocp.php>

Роб Хаббард [Rob Hubbard] – гений. Он был простым музыкантом-поденщиком, пока не решил освоить язык ассемблера 6502 для C64. Используя свои способности к музыке, он раскрыл мощь удивительного SID-чипа в C64, обнаружив множество компьютерных уловок и трюков, заставляющих этот 8-битный звуковой чип делать совершенно неожиданные вещи. Его зажигаемые темы для разных игр эпохи восьмидесятых стали легендарными. Я всегда любил Thrust (и не один я: ее потом записали в оркестровом исполнении), но спрос на темы был настолько велик, что выбирать есть из чего. Впрочем, вы и вы сами можете найти информацию о Хаббарде.

Если вам невдомек, какое отношение все это имеет к частоте вещания – успокойтесь, я как раз к этому перехожу. *Open Cubic Player* – это музыкальный плеер, который, помимо прочих форматов, поддерживает широко распространенный SID. Та-

ких файлов полно в сети, и они преданно хранят музыку той эпохи. *OCF* – возрожденный старый инструмент, поддерживающий этот формат (используя *libsndplay*) и прочие, столь же прекрасные, но выбивающиеся из основного русла форматы. На нем также воспроизводятся файлы MP3 и даже FLAC аудио.

Пользовательский интерфейс *OCF* основан на *Curses* – ах, этот прелестный дух истории! – и снабжен двухканальным дисплеем частоты, что также способствует созданию атмосферы восьмидесятых при воспроизведении SID-треков.

Предупреждение для пользователей SELinux: этот старый брюзга начнет жаловаться на библиотеки, включенные в *OCF*,

«Интерфейс OCF основан на Curses – ах, этот дух истории!»



► **Пой со мной, и мы мысленно вернемся в эпоху подземки. О, эта ностальгия...**

так что вам придется отключить его, или сделать следующее:

```
chcon -t texrel_shlib_t /usr/local/lib/ocp-0.1.20/autoload/*.so
```

Недостатки у *OCF* есть: он не любит делиться аудиоканалом и периодически отказывается выключаться. Но это лишь мелкие дефекты отличного инструмента, позволяющего вам совершить путешествие в музыкальное прошлое.

Инструмент ТВ -листингов

XMLTV

Версия 0.5.57 Сайт <http://wiki.xmltv.org>

Удивительно, но в некой разновидности примитивной международной сети многие компании конвертируют видео в цифровой сигнал и затем покрывают огромное расстояние, передавая эти сигналы со спутников по оптоволокну или с огромных ретрансляторных вышек, утыкавших все вокруг. Еще более удивительно, что, по крайней мере в Великобритании, вы им за это платите. А потом они не хотят говорить вам, что будет по ТВ.

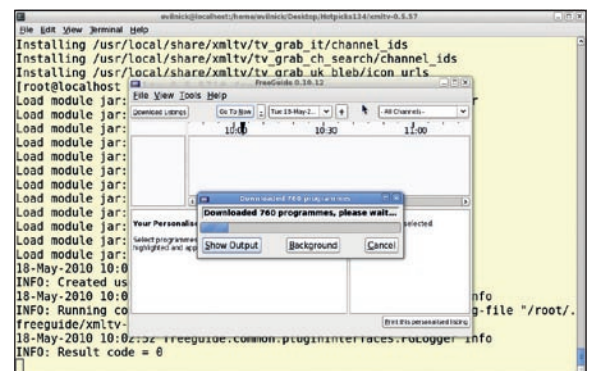
Есть масса причин, по которым вас может интересовать, что будет по телевидению в определенное время. Возможно, вам хочется записать эту передачу какой-нибудь хитрой PVR-системой; или – автоматически вносить в свой ежедневник основные шоу; или – отслеживать, сколько раз они повторяются. Или – всякое бывает – вы и правда хотите посмотреть телевизор...

XMLTV – не одиночное приложение. По сути, это скорее подборка инструментов,

работающих по общему протоколу. Неудивительно, что построено это все на XML-тексте, который одновременно прост(оват) в чтении и достаточно несложен для программирования.

Система использует огромный пакет захватывающих (в смысле, хватающих) скриптов-грабберов, которые и выполняют работу по сбору данных из разнообразных источников и перекодированию их в XMLTV-формат. Проблема грабберов в том, что работать им приходится с источниками, подверженными изменениям: например, с сайтами, предназначенными в основном для просмотра людьми, а не для автоматического извлечения информации.

«В Британии грабберы для ТВ подверглись модернизации.»



► **Многие приложения PVR и телегиды, например, Freeguide, используют движки XMLTV для работы с листингами.**

Однако в Великобритании грабберы для телевидения подверглись модернизации, вот мы и включили в наш материал *XMLTV*. Одно из улучшений – привязка к сервису BBC Backstage, так что будем надеяться на более точные результаты. Если для своих просмотров вы используете родственные проекты, обновление должно подойти гладко, но чтобы вполне насладиться радостями нового кода грабберов, вам, видимо, придется обновить конфигурацию (в зависимости от способа просмотра).

Предиктивный текст

Presage 0.8.2

Версия 0.8.2 Сайт <http://freshmeat.net/projects/presage>

Предиктивный текст – это чушь. В большинстве случаев, не так уж он предиктивен (предсказующий), потому что не пытается понять, о чем вы пишете, а просто автодополняет наиболее частыми словами, начинающимися с тех букв, что вы вводите. Это иногда экономит время, но может также и привести к тому, что вы случайно поведаете знакомым, что собираетесь «проехать по пустому туману», «загнуть куда-то по дороге домой», или что «отличный Филипп идет в кино». Короче говоря, отстой.

Presage – это не про автодополнение набираемых слов. Он любит размышлять о перспективе, поэтому предлагает вам целиком все слово, которое вы якобы наберете следующим. Если оно неправильное, он продолжает гадать, пока не попадет в точку, или пока вы не напечатаете слово полностью. *Presage* отслеживает то, что вы печатаете, или обрабатывает существующие текстовые файлы и создает

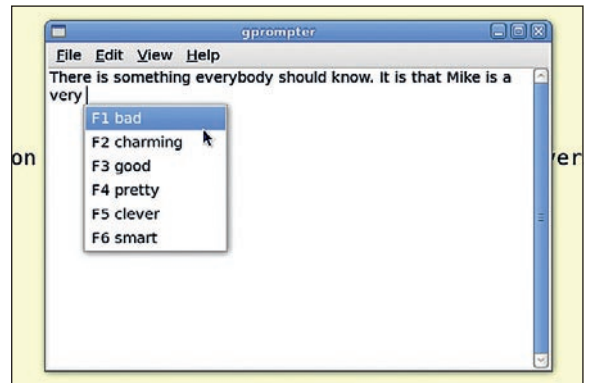
базу данных используемых слов, чтобы повысить эффективность прогноза.

Возможно, вы поторопитесь сбросить *Presage* со счетов, поскольку с ним взаимодействует ограниченное количество приложений. Подумайте еще раз. Одна из утилит в *Presage* – помощник для кросс-платформенного Assistive Technology Service Provider Interface (AT-SPI), который реализован в Gnome и поддерживается в KDE.

Широкий выбор

Теоретически, любое приложение, предусматривающее помощь при вводе людям с нарушением, скажем, зрительной функции, по специальным ИТ-технологиям,

«Presage предложит целиком слово, которое вы якобы набираете.»



› *Presage* отображает меню с подсказками вариантов следующего слова, хотя, если вы печатаете вслепую, вас это может затормозить.

обязано работать с *Presage*, если включена совместимость. Конечно, есть и несколько сопутствующих приложений для простейшей обработки текста.

Большинство примеров используют своего рода всплывающее меню, которое предлагает вам нажать на функциональную клавишу, чтобы завершить слово. К этому нужно привыкнуть; если вы печатаете быстро и точно, то это только замедлит темп вашей работы, но есть немало областей, где такое может очень пригодиться.

Менеджер паролей

KeePassX

Версия 0.4.3 Сайт www.keepassx.org

Пару лет назад я применял пароли на основе узоров, создаваемых ими на клавиатуре. Это была отличная система, дававшая надежные запоминающиеся пароли, которые нечего было надеяться взломать на основе словаря. Но увы – система не работала, если мне нужен был доступ к чему-либо через телефон.

Пароли – это необходимое зло. Хакеры былых времен из MIT [Массачусетского Технологического Института, – прим. пер.], возможно, считали их ненужным отвлекающим фактором (очевидно, RMS было потрачено немало времени на удаление разнообразных систем паролей в этом учреждении), но тогда еще не было кредитных карт и систем онлайн-покупок, хулиганов-скриптовщиков и обильной персональной информации, хранящейся под учетными записями в Интернете. Проблема в том, что иметь один пароль на все случаи жизни – значит, нарваться на неприятности: злоумышленники давно знают, что у большин-

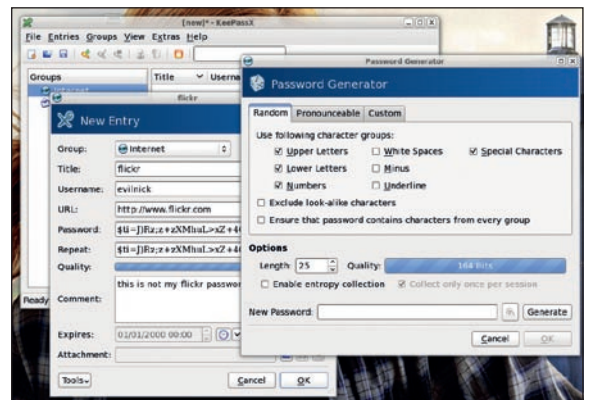
ства ситуация именно такова, и добыв один пароль, наносят вам серьезный ущерб.

KeePassX имеет долгую историю. Появился на свет он в какой-то малоизвестной ОС, а теперь его внедрили в OS X и Linux.

Зашифрованная база

Концепция проста: это база данных, где хранятся ваши пароли. Естественно, сама база зашифрована (с помощью Advanced Encryption Standard или Twofish), и чтобы открыть ее, нужен пароль или файл ключа. Или и то, и другое. Это интересная разработка, потому что она позволяет добавить лишний уровень безопасности. Например, файл ключа может храниться на USB-брелочке.

«KWallet хорош, но у KeePassX есть дополнительные функции.»



› *KeePassX* позволяет указать, что включить в пароль, а затем генерирует легкие для запоминания пароли.

Конечно, мы все привыкли к радостям *KWallet*, так зачем нам другой менеджер паролей? Что ж, *KWallet* отлично подходит для многих задач, но у *KeePassX* есть весьма привлекательные дополнительные функции. Вы можете добавлять в пароль контекстуальные данные – приложить URL, добавить примечания, сохранить имя пользователя; можно даже приложить произвольно выбранный файл. *KeePassX* также помогает сгенерировать полезные пароли в указанных пользователем параметрах и следит за сроком действия пароля. Возможно, в будущем нас ждет биометрика по всему телу, но на данный момент сойдет и это.

HotGames Развлекательные приложения

Сортировщик фигур

Tubularix

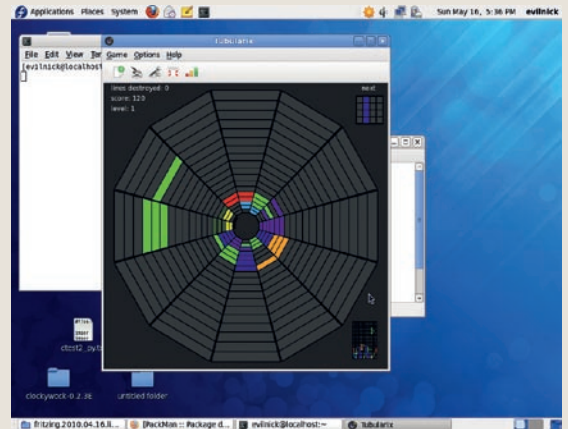
Версия 0.4.0.8 Сайт <http://tubularix.sourceforge.net>

Представьте себе мир без стен. Хм-м. Нет, не так. Представьте себе мир, где есть одна огромная стена, опоясывающая его по окружности, подобно гигантскому цилиндру (вспомните здание Понте в Йоханнесбурге). Это стена, у которой нет краев. И если вы используете ее в качестве игровой поверхности в игре типа *Tetris*, все будет заворачиваться по кругу и создавать постоянную путаницу. Так что добро пожаловать в запутанный и скругленный мир *Tubularix* – закрученное ответвление *Tempest* и *Tetris* (возможно, следовало бы дать ей название *Temptris*). Иисусе! Вдруг это читают юристы, занимающиеся авторским правом – в таком случае оно невероятно оригинально, это просто шедевр, отличающийся крайней глубиной, мастерством и талантом.

Вы, должно быть, уже ухватили суть. Глядя в трубу, в которую падают странно искривленные, но знакомые блоки, вы клавишами курсора перемещаете их по цилиндру, не теряя надежду расположить их в виде концентрических окружностей на дне. Ели это произойдет, линия исчезнет, все сместится вниз и вы получите награду в виде количества очков, которое кажется случайным числом – ура, все победили.

Однако все не так просто, как кажется. Возможно, дело в отсутствии границ (что в теории должно облегчить си-

«Добро пожаловать в запутанный и скругленный мир Tubularix.»



» То, что вы набили руку в *Tetris*, вовсе не означает, что и в *Tubularix* будет так же. Поверьте, не будет.

туацию, поскольку для каждой фигуры больше удачных позиций) или в сложности приспособления к изломанной форме фигур; но это труднее, чем можно ожидать. Возможно, именно поэтому здание Понте известно высоким уровнем преступности.

Эта версия доступна в виде RPM на сайте, либо вы можете собрать ее из исходников, если вам такое нравится; тогда понадобится *Qt 4* и тщательное выполнение инструкций – вам придется дважды запускать *qmake*.

Настольная игра

Pasang Emas

Версия 2.1.0 Сайт pasang-emas.sourceforge.net

Вы на дружеской ноге с султаном Брунея и даже входите на его день рождения? Среди прочих традиций, связанных с этим днем (очевидно, помимо покупки для султана новой машины и торта) – игра в ставшую всемирно известной *Pasang Emas*. Ну, я-то с султанами не дружу. Но я рад, что узнал об этой игре, потому что она хорошая.

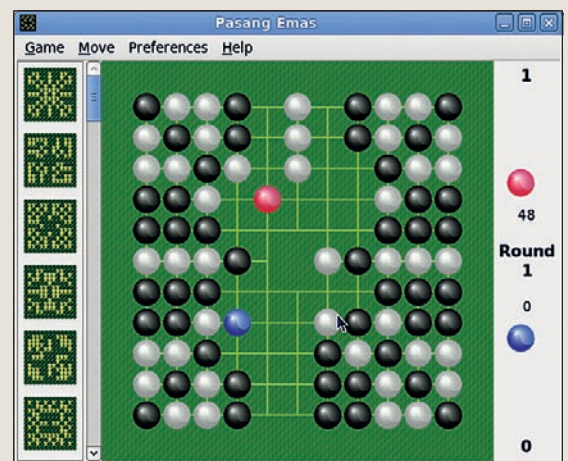
Она несколько напоминает *Solitaire* – все те же цветные камушки в клеточках, но чертовски более сложна. Два игрока по очереди перепрыгивают через непрерывные ряды фишек. Количество должно быть нечетным. Захваченные фишки приносят одно или два очка, в зависимости от их цвета. Фишки игроков представлены красным или голубым шариком, которые входят в игру, как только вы создадите проход. На сайте объясня-

ются правила, и над ними придется задуматься, но следовать им на практике очень легко.

Одна из хитростей игры – узоры камушков в самом начале. Предполагается, что они могут в корне изменить стратегию игры, но они генерируются псевдослучайным образом (очевидно, традиционные узоры взяты из мотивов вышивки).

Я – новичок, но предупреждаю, что надо избегать позорной ситуации под названием «suntuk», т.е., неспособности сделать допустимый ход, когда на поле еще есть фишки. Если такое

«Одна из хитростей игры — узоры из фишек в самом начале.»



» Следите, как бы не оказаться в положении, когда нельзя сделать следующий ход.

случается, вы теряете все свои фишки и немедленно штрафуетесь. Ужас.

Чтобы собрать эту игру из исходников, вам, помимо прочего, понадобится *gnome-doc-utils*. Во всем остальном она компилируется и играется без проблем. Но чтобы победить, нужно немало времени.

Схемотехника

Fritzing

Версия 0.3.19b Сайт <http://fritzing.org>

Вас тоже заразило безумие Arduino? Раскошелились на оборудование и создавали разные наброски? Которые так и не реализовались? Вы решили, что проектирование — чересчур кропотливая работа? А может, вы спроектировали десятки вещей, и даже реализовали их. Может быть, вам даже удалось заставить их работать. Но когда вы захотели поделиться своими созданиями с другими, оказалось, что под рукой нет простого инструмента, способного вам помочь.

Во всех этих случаях вам стоит обратиться к *Fritzing*. Созданный в качестве проекта Университетом прикладных наук (University of Applied Science) в Потсдаме, *Fritzing* являет собой инструмент для содействия в сборке Arduino или иных простых экспериментов над макетными платами, начиная с фазы тестирования и до создания полного дизайна печатной платы.

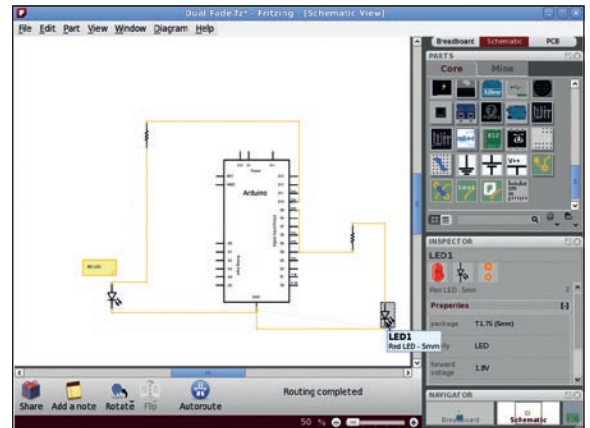
Есть три разных вида. В идеале вы начинаете создание своего прототипа в режиме макетной платы. Цепляйте и перетаскивайте компоненты с панели справа и размещайте их в нужных местах на плате. Компоненты моделируются графически и физически, чтобы они выглядели реалистично и вы не создали модель, не во-

площаемую на практике. В основной библиотеке полно компонентов, и вы найдете все, что вам нужно. Помимо стандартных пассивных компонентов, есть немало других вещиц из волшебного ящика экспериментатора: например, модули микрофона, выключатели, срабатывающие при наклоне, 7-сегментные дисплеи и т.п. «Просто микросхемы» могут быть представлены компонентом IC, при этом какие-то специфические детали можно добавить в свойства компонента.

Этот редактор схем настолько хитроумен, что даже предупредит вас, если вы сделаете глупость: например, коротко замкнете выводы питания.

Больше способов правки

Нажмите на вкладку запуска редактора схем, и перед вами, несомненно, предстанет суцья мешанина. Все компоненты изображены стандартными символами электросхем, но сперва провода между ними представлены простым соединением одной точки с другой. Нажмите на инструмент Autoroute, и программа рассортирует все это для вас. Или, по крайней мере, честно попытается. Действия эти не отличаются совершенством и быстротой, но демонстрируют рвение все упоря-

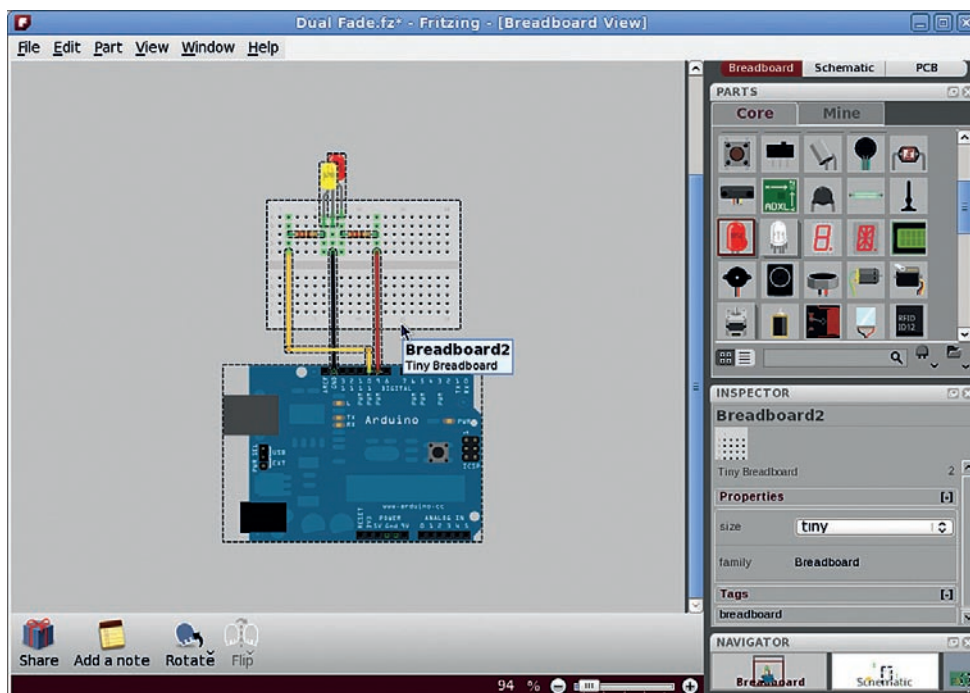


» Добавьте всего пару компонентов к нашему дизайну, и получится весьма реалистичный имитатор Майка.

дочить до познаваемого вида. Конечно, можно отредактировать расположение объектов и тем помочь программе. Примечания к дизайну добавлены небольшими гаджетами в стиле «желтых листочков», которые можно перетаскивать в самые разные места.

Конечный режим просмотра — схема печатной платы. Помимо детализации внешнего вида компонентов, библиотека включает габариты и знает, насколько далеко друг от друга располагать отверстия для монтажа и т.п. Что касается схемы, то при взгляде на нее она может показаться полной неразберихой. Здесь инструмент Autoroute работает еще мед-

«Хитроумный редактор предупредит вас, если вы сделали глупость.»



» К вашему дизайну можно добавлять элементы из одной из внешних библиотек *Fritzing*.

леннее и еще больше подвержен ошибкам; но очень забавно следовать за ним по мере его продвижения по разным углам и пытаться угадать, как они должны размещаться в пространстве и т.д. И снова, некоторое содействие человеческого мозга не будет здесь лишним.

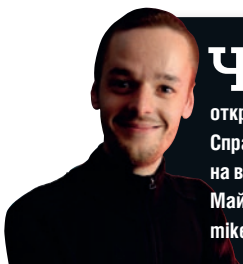
Имеется опция для экспорта вашей схемы тротровки платы в PDF, но на данной стадии разработки она, возможно, ни к чему.

Fritzing пока находится на альфа-стадии, но вполне способен составлять схему расположения компонентов. Стратегический план — интеграция способностей составлять схему соединений с другими программами (типа вездесущего редактора схем *Eagle*) и исправление интерфейса пользователя, но в перспективе он также должен включать в себя некий имитатор Arduino.

Честно говоря, во *Fritzing* много чего не хватает, но проект имеет солидную движущую силу; немалую лепту вносят пользователи, а также продолжается разработка парнями из Потсдама.

На диске

Масса дистрибутивов и программ для ознакомления...



Читайте, чтобы найти всю информацию, которая вам понадобится для использования этого DVD! Если вы новичок в Linux, откройте `index.html` на диске (Сторона 1) и перейдите в раздел Справка: там вы найдете мини-учебник, дающий ответы на вопросы 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

Дистрибутив Linux

Linux Mint 9

Linux Mint сейчас занимает третье место в рейтинге популярности DistroWatch, и вполне заслуженно: это классный дистрибутив, который расширил свою основу Ubuntu отличным набором программ, свежими темами и новыми инструментами настройки. Mint 9 основан на Ubuntu 10.04, релизе LTS (Long Term Support – с долгосрочной поддержкой), и работает непосредственно с нашего DVD с двойной загрузкой в режиме Live, так что вам не придется устанавливать его на свой жесткий диск, если вы захотите его попробовать.

Системные требования, рекомендуемые для нормальной работы, таковы:

- » 1,5 ГГц CPU
- » 512 МБ ОЗУ
- » 10 ГБ свободного места на жестком диске

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.

Версия Mint 9 на LXF DVD 32-битная, но отлично ладит и с 64-битными машинами. Для ее запуска загрузите компьютер с DVD; если ваша машина настроена на загрузку с жесткого диска, нужно будет изменить порядок загрузки в BIOS. Обычно это делается нажатием на клавишу F2, Esc или Del во время старта компьютера – текст на экране подскажет вам, что делать.

В меню загрузки DVD нажимайте на стрелку вниз и на Enter, чтобы выбрать Mint 9, и он загрузится. Если возникнут проблемы, попробуйте перезагрузиться и выбрать режим совместимости (Compatibility mode) в загрузочном меню.

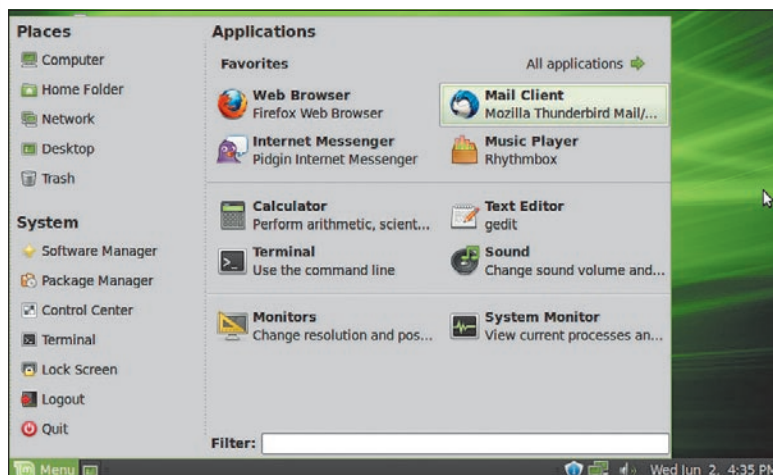
При появлении рабочего стола нажмите на кнопку Menu [Меню] в нижнем левом углу, чтобы ознакомиться с имеющимися приложениями. По умолчанию отображаются значки самых популярных программ, но вы можете увидеть весь список, нажав на All applications [Все приложения] в правом верхнем углу. В подменю Administration [Администрирование] и Preferences [Настройки] находятся программы для настройки системы – например, для установки дополнительных программ из Интернета, управления учетными записями пользователей и т. д.

Установка Mint

Если вам понравился Mint и вы хотите поселить его у себя на жестком диске, дважды щелкните по значку Install [Установить] на рабочем столе. Вам предложат выбрать язык и часовой пояс, а затем разбить жесткий диск на разделы. При желании установить Mint на весь диск целиком, выберите опцию «Erase And Use Entire Disk» (LXF133).

Если хотите, чтобы установка была совместима с Windows, можете изменить размер раздела Windows на диске, чтобы выделить место. (Неплохо было бы сначала сделать резервные копии важных данных в Windows, так, на всякий случай!) После установки появится загрузочное меню, в котором вы будете выбирать нужную ОС.

Программа установки предложит вам выбрать имя пользователя и пароль – не забывайте их! После этого загрузится вновь установленный Mint. Более подробная информация имеется на www.linuxmint.com, а если понадобится помощь, вы сможете получить ее от других пользователей Mint на <http://forums.linuxmint.com> (англ.) или на форуме русскоязычного сообщества: <http://mintlinux.ru>.



» Linux Mint: освежит ваш ПК, естественно!

Операционные системы

Mepis 8.5, Puppy, System Rescue, Haiku

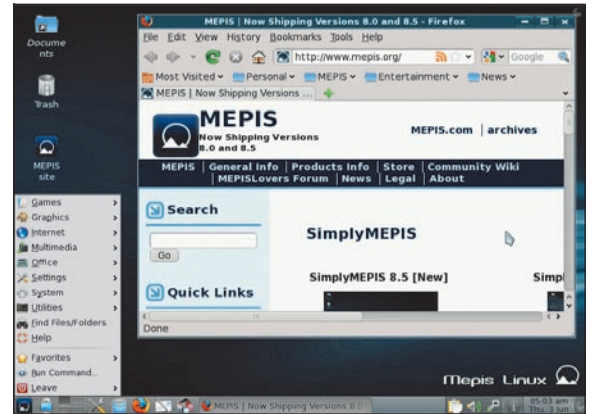
Mepis (www.mepis.org) – хороший всесторонний дистрибутив, особенно подходящий для начинающих линуксоидов. Новый релиз 8.5 появился этой весной, и вы найдете его в разделе Дистрибутивы (Distros) нашего DVD в формате ISO. Это – образ диска для записи на CD-R; помните, что вы должны записать его именно как образ ISO, а не просто скопировать файл на пустой диск. Если вы не вполне уверены в том, как это сделать, обратитесь к документации вашей программы для записи дисков.

Записав диск, запустите с него ваш компьютер. В загрузочном меню выберите опцию по умолчанию, и вскоре появится экран приглашения. Вы можете войти под именем пользователя «demo» с паролем «demo». (Пароль для root, или администратора – «root».) На рабочем столе можно познакомиться с наличными программами,

нажав на кнопку в нижнем левом углу. Чтобы выйти из системы, нажмите на ту же кнопку и выберите Leave [Выйти]. Если вы захотите установить Mepis на жесткий диск, дважды щелкните по значку Install на рабочем столе и следуйте подсказкам.

Puppy и System Rescue – компактные дистрибутивы, созданные для маломощных машин и для восстановления системы. Если вы хотите реанимировать старый Pentium II со 128 МБ ОЗУ, запишите ISO Puppy на CD-R и загрузитесь с него – это отличная настольная мини-ОС. CD System Rescue, также имеющийся в виде ISO, стоит постоянно иметь под рукой, поскольку в ней имеются инструменты для восстановления системы в случае проблем.

И, наконец, в разделе Дистрибутивы (Distros) у нас есть самый свежий альфа-релиз Haiku, настольной операционной системы, испытывающей сильнейшее вли-



» Mepis, возможно, мрачноват на вид, зато дружелюбен к начинающим пользователям Linux.

яние BeOS. Haiku стремится стать быстрой, логичной и простой в использовании, и хотя достижения средней стадии разработки пока что грешат ошибками, они демонстрируют хороший рывок вперед. Вы найдете ее в виде образа «anyboot» – его можно записать на CD-R или на USB-брелок; подробную информацию вы найдете в файле README.TXT на диске.

Другие программы

UndoDB, Synfig, podcasts

На стр. 12 рассказывается об UndoDB, «обратном» отладчике, и мы решили, что было бы обидно не дать вам попробовать этот инструмент в деле. В разделе Разработка [Development] вашего DVD имеется пробная версия коммерческого издания. Срок ее действия истекает 24 сентября 2010 года, и вам хватит времени, чтобы ее оценить. Кто не намерен использовать UndoDB в коммерческих целях, может запросить бесплатную редакцию Starter с сайта http://undo-software.com/undodb_requestnoncommercial.html. В ней нет ряда продвинутых функций (например, отладки программ, использующих разделяемую память), но она вполне годна для работы над свободным или открытым проектом.

А в разделе Рабочий стол [Desktop] у нас самый свежий релиз Synfig Studio (0.62.01), мощного векторного пакета для анимации. Это один из самых продуманных и профессиональных проектов с открытым кодом из нам известных, и разработчики заслуживают высочайших похвал за свои труды. Вдобавок он реально прост: команда Synfig выдает готовые к работе двоичные пакеты. Сохраните Deb-файл на рабочем столе

и дважды щелкните на нем в Debian/Ubuntu, или, если у вас Fedora/Mandriva/OpenSUSE, поступите так же с RPM-пакетом. Для других дистрибутивов есть универсальный бинарник .tar.bz2, а также 64-битные версии.

Теперь о стандартных рубриках – Ответы и Новичку в Linux; обе сидят в разделе Справка (Help). Если вы хотите спросить у нас, как справиться с вашей проблемой в Linux, сперва загляните в Ответы: там собрано несколько сотен обращений в нашу службу поддержки, и велик шанс, что ваша проблема уже решена. А если вы только приступили к знакомству с Linux и не понимаете каких-либо терминов, просмотрите наше краткое введение для новичков.



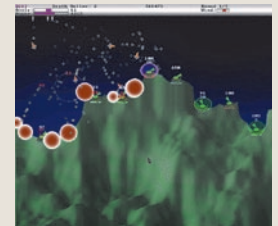
» Synfig Studio: профессиональная векторная 2D-анимация – причем даром.

И более того! Пять классных игр

После всех этих серьезных дистрибутивов и программ, расслабьтесь в нашем разделе Игр [Games]. У нас есть демо-версия *Osmos* – игры, способной вмешаться в ваши сны; есть Deb-пакет для Ubuntu/Debian и RPM для Fedora, Mandriva и OpenSUSE. Если у вас другой дистрибутив, распакуйте архив .tar.gz и прочитайте в нем файл `readme.html`.

Далее по списку идет *Armagetron Advanced*, версия игры *Armagetron*, расширенная такими функциями, как индикация на лобовом стекле. Это – игра *Trojan*, снова и снова: направляйте свой светочик по арене, оставляя за собой след, и старайтесь сами не попасть в свой след или в след других игроков!

Atomic Tanks – очень забавная игра в стиле *Worms*. Еще



» Atomic Tanks: когда взрывы – это круто.

у нас есть *Bombic 2*, в стиле *Bombberman*, где игроки мечутся по 2D-арене, закладывая бомбы с часовым механизмом в попытке победить оппозицию. У вас может быть до четырех игроков на одной машине. И, наконец, *LentilWars* – сюрреалистический «настольный конфликт», в котором Smarties [шоколадное драже, – прим. пер.] сражаются с m&Ms. Примило!

На диске

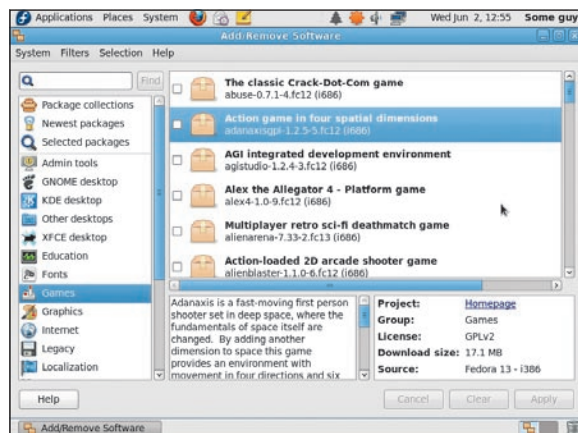
Побалуйте свой Linux-компьютер свежим ПО

Дистрибутив Linux

Fedora 13

Релиз подзадержался на пару недель из-за исправления ошибок, обнаруженных в последний момент; и вот Fedora 13 с нами. Как всегда, он идет в ногу с последними достижениями в области ПО и технологий Linux и дает наилучшую возможность узнать, что день грядущий готовит для Linux. А если вы прежде не использовали Fedora, не робейте: находясь на передовых позициях, он не вторгается на территорию нестабильности, так что это – отличный дистрибутив, как для дома, так и для рабочей станции. Более подробная информация о Fedora приводится на стр. 30, а здесь мы опишем процесс его установки.

➤ Тысячи дополнительных программ можно найти в Интернете – нажмите на **System > Administration > Add/Remove Software** для их просмотра.



Fedora 13 загружается с LXF DVD в режиме Live, и вы сможете познакомиться с ним без инсталляции на жесткий диск. Ясно, что работа с оптическим носителем медленнее, чем с жесткого диска, и если вы хотите прописать Fedora постоянно, следуйте указаниям внизу этой страницы. Версия Fedora 13 на диске 32-битная, но поладит и с 64-битными машинами (x86-64).

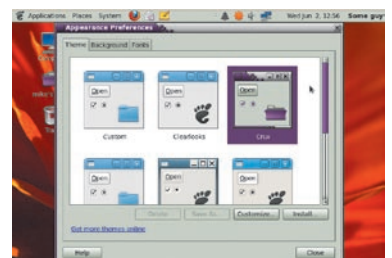
Минимальные системные требования:

- 400 МГц Intel/AMD (или совместимый) процессор.
- 1 ГБ ОЗУ.
- 10 ГБ на жестком диске.

Помните, что 400 МГц – это минимальные требования для сервера; есть шансы, что Fedora будет у вас настольной графической ОС, и тогда мы рекомендуем для нормальной работы процессор на 1,5 ГГц или выше. Fedora здорово умеет распознавать оборудование, и ваши периферийные устройства должны сразу же заработать, но если с загрузкой дистрибутива будут проблемы, попробуйте удалить ненужные устройства – например, принтеры или сканеры, поскольку, возможно, ОС затрудняется с их определением.

Установка Fedora

Фаза разбивки диска на разделы требует максимума внимания при установке любой ОС. Как отдельная ОС, Fedora требует пер-



➤ Войдите в **System > Preferences > Appearance** для выбора разных тем виджетов и обоев рабочего стола.

сонального места на жестком диске. Если это ваш первый опыт использования Linux, то у вас, скорее всего, жесткий диск либо пуст, либо целиком отведен под Windows.

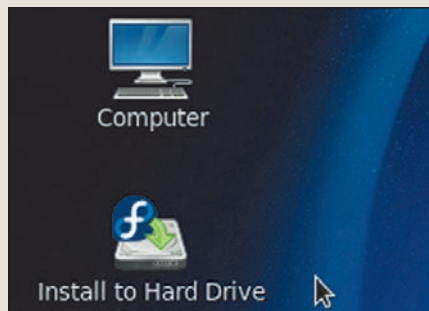
В первом случае можно указать установщику Fedora отвести весь диск под Linux – это сверхпросто. Если у вас Windows, сперва загрузите Windows, запустите дефрагментацию диска и сохраните важные файлы в надежном месте. Затем, устанавливая Fedora, вы сможете сжать раздел Windows на диске, чтобы отвести место под Linux. Этот процесс протекает без запинок в 99,9% случаев, но любая операция по изменению диска потенциально опасна для данных, поэтому сначала сделайте их резервные копии! После установки вы увидите меню загрузки, чтобы выбрать нужную вам ОС при загрузке компьютера.

Шаг за шагом: Устанавливаем Fedora 13



1 Загрузка

Запустите компьютер с LXF DVD и нажмите на Enter в загрузочном меню. (Возможно, вам придется изменить порядок загрузки в BIOS — в этом случае загляните в документацию своего компьютера).



2 Рабочий стол

В экране приглашения нажмите на Automatic Login, чтобы перейти на рабочий стол. Нажмите на меню Applications, чтобы просмотреть программы, и дважды щелкните по значку Install для установки.



3 Жесткий диск

Выберите язык, и затем появится этот экран с вопросом, на какой диск вы хотите установить Fedora. Отметьте нужный вариант и нажмите Next.

Как бы мне?..

Если вы – совершенный новичок в Linux, то после установки Fedora вы можете не знать, как выполнить определенные задачи. Вот список наиболее частых задач и способов их решения...

» **Работа в сети** Нажмите на Приложения > Интернет > Firefox, чтобы запустить второй по популярности в мире браузер, поддерживаемый сотнями расширений.

» **Работа с файлами** Откройте Переходы > Домашняя папка.

» **Воспроизведение видео и аудио** В подменю Звук и видео вы найдете музыкальный плеер *Rhythmbox* и пункт Видеопроигрыватель, который запускает *Totem*.

» **Риппинг и прожиг дисков** Также в подменю Звук и видео вы увидите Экстрактор аудио-CD, который позволяет конвертировать диски в аудиофайлы, и программу *Brasero* для записи CD/DVD-R.

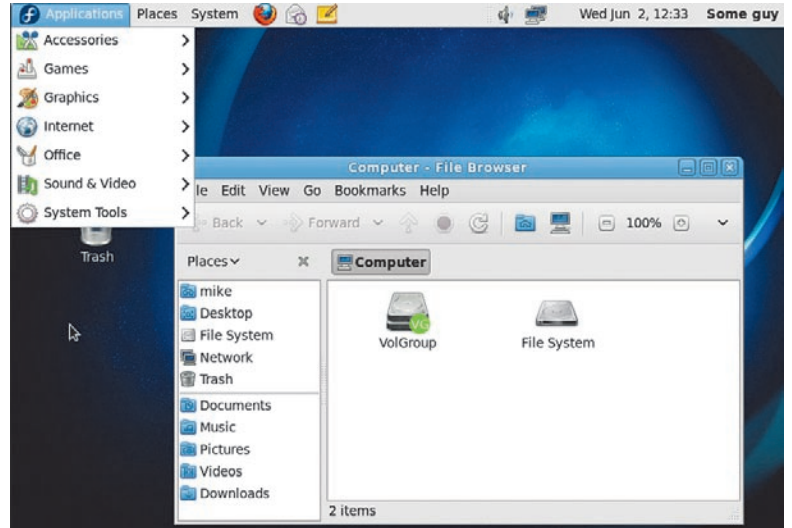
» **Чат онлайн** В меню Интернет найдите *Empathy*. Эта программа для обмена сообщениями поддерживает все популярные протоколы (MSN, AIM, Google и т.д.) Также есть *Pino* для Twitter.

» **Настройка системы** Нажмите на меню Система и перейдите в подменю Настройки, чтобы найти опции изменения рабочего стола, клавиатуры, настроек питания и сети, а в Администрировании вы сможете управлять пользователями, настраивать принтеры и получать обновления.

» **Устанавливать новые программы** Откройте Система > Администрирование > Установка и удаление программ, чтобы загружать программы из Интернета. Вы сможете просматривать категории программ, смотреть, насколько велика загрузка, и т.п.

» **Выключить** Нажмите на меню Система и затем на Завершение работы, чтобы перезагрузить или выключить компьютер.

Fedora имеет одно из самых деловых сообществ из всех дистрибутивов Linux, и сосредоточено оно вокруг главного сайта проекта www.fedoraproject.org. На нем есть ссылки на другие ресурсы,



сы, например, на <http://docs.fedoraproject.org> – нажмите на ссылку Fedora на левой панели, и затем на 13, чтобы раскрыть список руководств по этой версии дистрибутива.

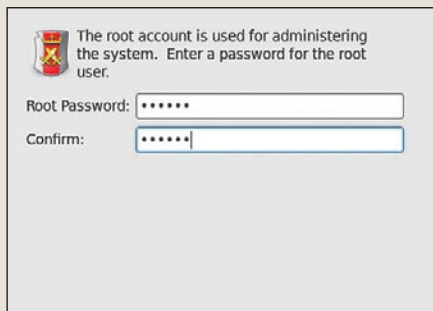
Если при установке или во время работы с Fedora у вас возникнут проблемы, посетите форум проекта Russian Fedora (<http://forum.russianfedora.ru>) и задайте свой вопрос. Другой хороший (англоязычный) ресурс – www.fedoraforum.org. **LXF**

» **Косметически Fedora 13 не очень далеко ушла от своей предыдущей версии, хотя некоторые значки новые.**

Не пропустите...

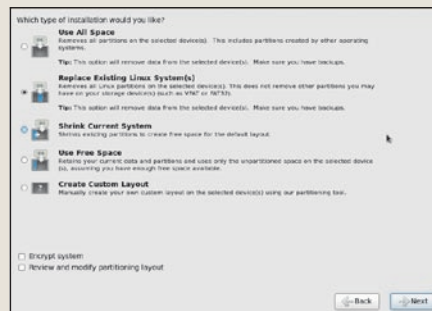
Shotwell
Прощай, *F-Spot* – новой программой по умолчанию для управления и работы с фото стал *Shotwell*.

Pino
Общайтесь со своими приятелями по Twitter и Identica с помощью *Pino*, нового клиента микроблоггинга.



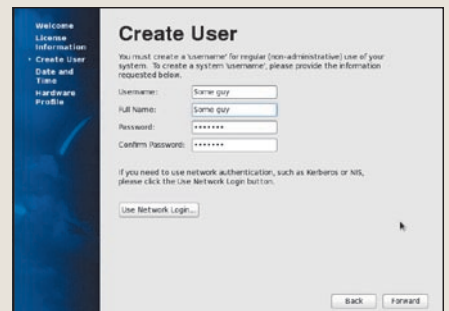
4 Пароль

Вам будет предложено задать имя для своего компьютера и выбрать местоположение и пароль для root (администратора). Не забудьте его! (В пароле важен регистр.)



5 Разбиение диска на разделы

Вы можете отвести весь диск под Linux, заменить имеющуюся установку Linux или сжать имеющуюся систему Windows, чтобы выделить место (если возможно).



6 Завершение

Когда файлы будут скопированы, Fedora перезагрузится с жесткого диска и предложит вам создать учетную запись обычного пользователя для повседневной работы.

СИСТЕМНЫЙ администратор

Клонировем Windows с помощью Symantec Ghost

Насколько неуязвима ваша беспроводная сеть?

Active Directory вместо рабочей группы

Настраиваем DSPAM – ваш личный спам-фильтр

Как спасти данные, если отказал жесткий диск

Модифицируем BIOS

Все ли возможности ClamAV вы используете?

Что важно знать об IP-телефонии

Админские сказки

www.SAMAG.ru

В «Системном администраторе» вы не прочтете о:

- котировках валют
- сплетнях
- погоде
- политике
- развлечениях



В вашем распоряжении:

- опыт лучших IT-специалистов
- новые идеи и полезные советы
- самые эффективные решения в области системного и сетевого администрирования



Подпишитесь сейчас!

Роспечатать – 20780, 81655
 Пресса России – 87836
 Online-подписка – www.linuxcenter.ru



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.

По этой программе учебное заведение получает:

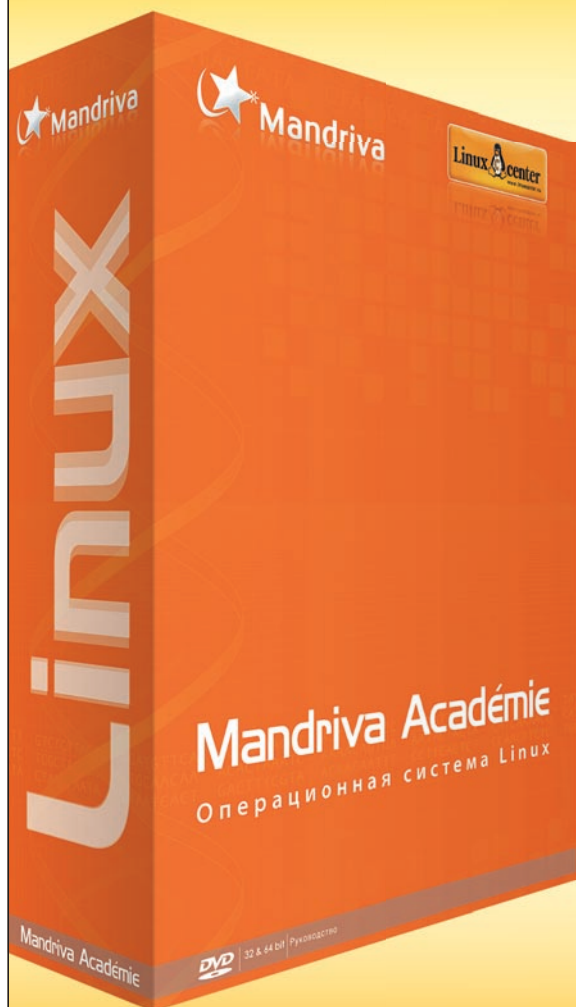
- ★ свежие версии дистрибутива Mandriva Linux (дважды в год)
- ★ доступ к обновлениям системы
- ★ техническую поддержку

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



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!

» Edimandriva 2010

» Альт Линукс 5.0 Школьный

» Ediphintu 9:10

ШКОЛЬНЫЙ Linux

Скоро 1 сентября — выберите себе дистрибутив для нового учебного года

Август 2010
LXF DVD 134

LINUX
FORMAT

fedora 13

Самый продвинутый в технологическом смысле дистрибутив — узнайте, как будет выглядеть Linux завтра

» Linux Mint 9
Популярный открытый Ubuntu озабочен новыми инструментами настройки

» Merp's 8.5

Легкий и дружелюбный к новичкам

» Наки

Если вам нужен Linux, попробуйте BeOS

А ТАКЖЕ: Purru Linux, MynthV 0.23, SSC 4.5

Август 2010
LXF DVD 134

LINUX
FORMAT

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:



LXF131
Май 2010

- » **Linux изнутри** Из каких компонентов состоит современный дистрибутив и как они взаимодействуют друг с другом.
- » **DOSBox** Лучший способ запустить DOS-программы в Linux.
- » **ClamAV** Свободный антивирус: руководство по эксплуатации.
- » **TaskJuggler** Управление проектами в свободном ПО.

LXFDVD: MOPSLinux 7.0 и Linux From Scratch 6.6

Печатная версия:
http://shop.linuxformat.ru/lxf_131/
Электронная версия в формате PDF:
http://shop.linuxformat.ru/elxf_131/



LXF132
Июнь 2010

- » **Пробуй любой дистрибутив!** Ваше пошаговое руководство по виртуализации в Linux.
- » **ARM** Обзор архитектуры, приобретающей все большую популярность.
- » **Живой звук** Создайте себе собственную систему реального времени.
- » **Файлы журналов** Где искать информацию, если что-то идет не так.

LXFDVD: PC-BSD 8.0, OpenClipart 2.0 и FlightGear 2.0

Печатная версия:
http://shop.linuxformat.ru/lxf_132/
Электронная версия в формате PDF:
http://shop.linuxformat.ru/elxf_132/



LXF133
Июль 2010

- » **Двойная загрузка** Правильный путь к мирному сосуществованию двух ОС.
- » **3D-редакторы** Сравним пять основных инструментов, доступных для Linux.
- » **XZip** Тестируем новый метод сжатия данных.
- » **Маршрутизация по политикам** Неизвестные особенности ядра Linux.

LXFDVD: Ubuntu 10.04 LTS LXF Edition, Sabayon 5.3 KDE & GNOME

Печатная версия:
http://shop.linuxformat.ru/lxf_133/
Электронная версия в формате PDF:
http://shop.linuxformat.ru/elxf_133/

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 получают электронную версию в подарок!

Спешите на www.linuxformat.ru/subscribe/

Специальное предложение!

Январь–июнь 2009

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски LXF. Если вы тоже озадачены этой проблемой, мы рады сообщить, что в интернет-магазине ГНУ/Линуксцентра продолжается распродажа журналов за 2009 год! Вы можете приобрести подшивки LXF за шесть или двенадцать месяцев со скидкой 30%.

Спешите — журналов осталось не так уж много!

Июль–декабрь 2009

http://shop.linuxformat.ru/lxf2009_01_06/

Январь–декабрь 2009

http://shop.linuxformat.ru/linuxformat_2009/

http://shop.linuxformat.ru/lxf2009_06_12/

Школа LXF

Спонсор рубрики
PingWin Software!
Созданная в мае 2009 года компания занимается поддержкой свободных продуктов, сообществ их разработчиков, пользователей и внедренцев.
www.pingwinsoft.ru

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Куда учителю податься?



По просьбам наших читателей, **Сергей Голубев** представляет очередное ежегодное сравнение «школьных» дистрибутивов.



Наш эксперт

Сергей Голубев
Член Общественного наблюдательного совета «школьного» проекта. Тестировал дистрибутивы, входящие в основную и расширенную репозитории.

У дорогих экипажей, полученных в виде подарка от добрых фей, есть одно неприятное свойство: ровно в полночь они имеют обыкновение превращаться в тыквы. Карета «Первой ПОмощи» – не исключение. В новогоднюю ночь, ровно в тот самый момент, когда часы на Спасской башне пробьют двенадцать раз, легальное приростное ПО, установленное во многих школах, превратится в контрафактное.

Сами же школы будут поставлены перед непростым выбором: заплатить денег и продлить лицензию или использовать свободное ПО. Причем, указанные выше сроки – сугубо теоретические. На практике же в учебных заведениях новый год начинается не 1 января, а 1 сентября. Конечно, можно сменить учебные планы и за время зимних каникул, но это будет непросто.

Таким образом, будет разумно предположить, что именно сейчас во многих учебных заведениях будет принято решение о переходе на СПО. Но на какое именно?

На сайте поддержки проекта www.spohelp.ru предлагается несколько вариантов: Edubuntu 9.10, Edumandriva 2010, Альт Линукс 5.0 Школьный, НауЛинукс 5.4. Этим список не исчерпывается – пользователям предлагаются также старые версии дистрибутива НауЛинукс и «просто Линукс», созданный в рамках специального проекта Министерства образования. О них мы сейчас говорить не будем, желая идти в ногу со временем и предполагая, что актуальные версии как минимум не хуже.

Цель обзора исключительно прагматична: помочь работникам школ сэкономить время, которое им придется потратить на выбор дистрибутива. Поэтому никаких теорий тут не будет – только практика на примере решения конкретных задач.

Критерии тестирования

Полное тестирование по всем школьным, окошкольным и внешкольным задачам по силам разве что серьезному институту. Поэтому мы даже и пытаться не будем, а сосредоточимся на выполнимом.

Тестовая машина – ноутбук Toshiba Satellite A200-1HV с видеоадаптером NVIDIA и беспроводной картой Intel Wireless WiFi Link 4965AGN. Не слишком старая, но и не очень новая.

В процессе тестирования будут рассматриваться следующие аспекты:

- » Наличие и качество работы автоматического режима установки (Windows-то нам хранить незачем – он уже фактически «просрочен»).
- » Выход в сеть через беспроводной 3G-модем связи Merlin XU870 (Интернет нужен всегда и везде).
- » Установка браузера *Chrome* и подключение *DropBox* (посмотрим, как система работает с приложениями сторонних авторов).
- » Просмотр видеоролика и прослушивание музыки (не будет ли проблем с кодеками?).
- » Удобство программирования на FreePascal.
- » Качество техподдержки.

Как вы заметили, сугубо школьных задач там всего ничего. И это неспроста. Одно из главных достоинств свободного ПО заключается в том, что его можно установить не только на учебные, но и на домашние машины – как сотрудников школ, так и учеников, причем совершенно бесплатно. Верно и обратное: если система не приживется дома, то и ее использование в школах будет проблематично (популярность Windows, кстати, не в последнюю очередь обусловлена ее универсальностью).

Подшивка LXF со скидкой 30%

стр. 107

Наконец, последнее замечание. Эту статью ни в коем случае нельзя рассматривать в качестве учебного пособия – автор не ставил перед собой такой цели. Единственная задача обзора – помочь учителю сделать оптимальный выбор в условиях крайне ограниченного времени на изучение системы.

Edubuntu 9.10

Установка системы производится несколькими щелчками мыши. При этом в автоматическом режиме сделать что-то неправильно практически невозможно. Даже стартовое меню выбора языка программы инсталляции появляется на экране само – нет нужды вызывать его нажатием какой-нибудь кнопки.

Сразу после установки система предложит пользователю обновиться до новой версии 10.04. Можно согласиться, а можно и отказаться – никакой принудилки тут нет.

Однако процедура обновления длится несколько часов (точное время зависит от «толщины» канала), поэтому нет особого смысла загружать предыдущую версию дистрибутива, если есть твердое намерение использовать самую новую. В пользу этого решения говорит то, что Ubuntu 10.04 – LTS-дистрибутив, обновления для которого будут поставляться в течение трёх лет. По мнению разработчиков, подобная схема делает продукт более привлекательным для корпоративного сектора, который предпочитает «долгоиграющие» решения.

Очевидно, что школы попадают под это определение.

Первичная настройка производится в полуавтоматическом режиме. Система предложит доустановить пакеты поддержки русского языка – соглашайтесь. Единственное, что потребовалось сделать «руками» – назначить удобную комбинацию клавиш для переключения раскладок.

Интересно то, что Ubuntu сама подстраивается под нужды пользователя. Допустим, захотел он включить визуальные эффекты – драйвер поддержки видеоадаптера установится автоматически. Это очень удобно, поскольку позволяет значительно сэкономить время.

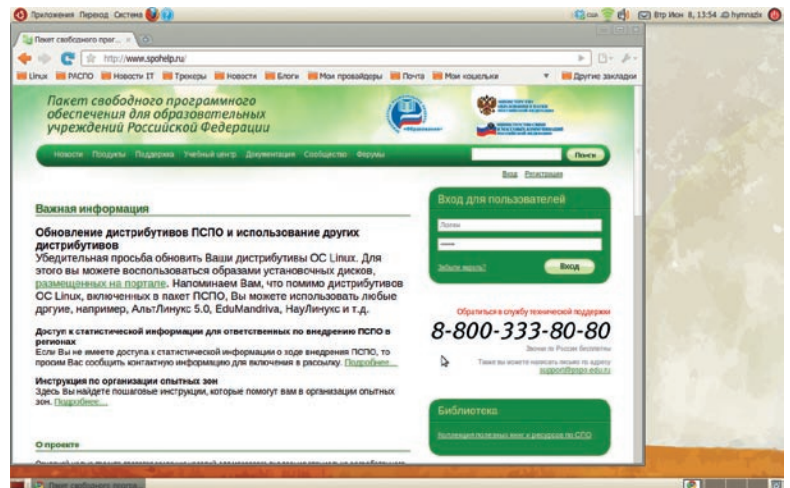
Наряду с этим система не ограничивает опытного пользователя, предпочитающего обходиться без автоматике. Например, можно самостоятельно скачать с официального сайта нужный драйвер NVIDIA и установить его запуском скрипта – в сети уже есть видеoinструкция на сей счет.

Таким образом, установка системы и ее первичная настройка осуществляется максимально удобным для неподготовленного пользователя способом. Эти операции занимают немного времени, и можно быстро переходить к каким-то практическим действиям.

Первое – выход в сеть через модем Merlin XU870. Устройство определилось автоматически – установки дополнительных драйверов не потребовалось. Осталось только войти в менеджер сетевых соединений и указать имя провайдера. Не потребовалось даже вводить PIN-код SIM-карты (если учесть, что его мало кто помнит, то это правильное решение разработчиков). Связь устойчива – работать можно.

Установка программ сторонних производителей также не вызывает затруднений. И для браузера *Chrome*, и для сервиса *DropBox* есть версии, предназначенные специально для Ubuntu. Пользователю надо только скачать deb-пакет и двойным щелчком запустить процесс инсталляции. По сути, та же процедура, что и в Windows – не легче и не сложнее. В результате пользователь может получить комфортную рабочую среду, ни в чем себе особо не отказывая и не ограничиваясь предложениями официального репозитория.

Воспроизведение мультимедийного контента требует кодеков, которые нельзя устанавливать автоматически, поскольку это мо-



жет противоречить законам некоторых стран. Однако пользователю Edubuntu не придется искать в сети нужные ему пакеты. Как только он захочет посмотреть видеоролик или прослушать звуковой файл, система спросит у него разрешения на инсталляцию требуемых модулей. Что-то более удобное сложно придумать даже теоретически.

Несмотря на то, что в состав системы по умолчанию входит много учебного ПО (есть даже знаменитая «черепашка»), FreePascal придется устанавливать дополнительно. Самый

простой способ – открыть «Центр приложений» и выбрать Lazarus, свободную среду разработки программного обеспечения для компилятора Free Pascal Compiler. Сам компилятор при этом установится автоматически.

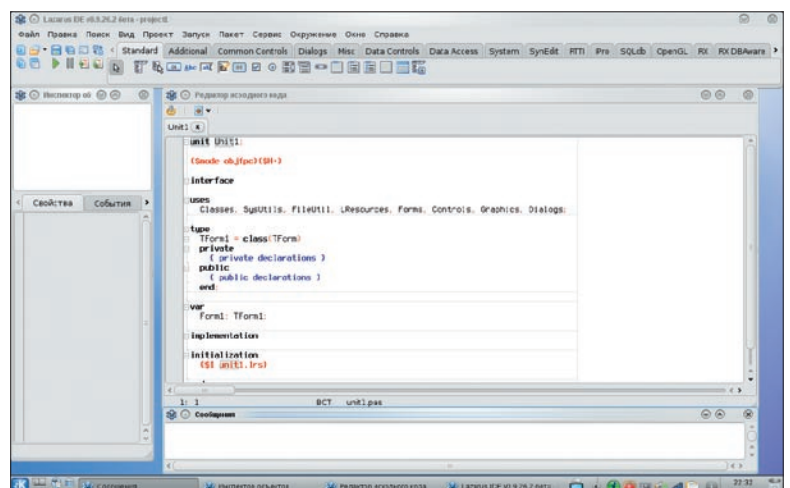
Если надо оболочку попроще, то неплохой выбор – Geany. Она также есть в официальном репозитории и устанавливается через «Центр приложений».

Таким образом, в кабинетах информатики Edubuntu будет чувствовать себя как дома. Причем, как нетрудно заметить, учителю не придется искать какие-то программы, покупать дополнительно какие-то диски – в дистрибутиве все есть. Правда, трафика уйдет изрядно. Но ведь все школы уже давно подключены к Сети, не правда ли?

Наконец, о технической поддержке. Ubuntu вообще (и Edubuntu в частности) отличается тем, что при решении каких-то простых

» Хотите попробовать браузер Chrome? С Edubuntu — нет проблем.

«В кабинетах информатики Edubuntu чувствует себя как дома.»



» Среда программирования Lazarus входит в состав Альт Линукс Школьный по умолчанию.

задач можно запросто обойтись без платных услуг. У этого дистрибутива очень мощное и активное сообщество, поэтому решение проблемы зачастую может быть найдено через поисковую систему. Кстати, значительная часть запросов, где фигурирует слово «Linux», приводят на сайты, где советы даются прежде всего пользователям Ubuntu.

Ну, а в сложных ситуациях можно воспользоваться либо технической поддержкой на портале www.spohelp.ru, либо заключить прямой договор с компанией PingWin Software, оказывающей такие услуги.

Резюме. Edubuntu – универсальная система, которая подходит как для школы, так и для дома. Все работает «само», никаких инструкций читать не пришлось, не говоря уже о поисках на форумах и тому подобных вещах.

Альт Линукс 5.0 Школьный

Установка системы в автоматическом режиме проходит быстро и довольно корректно. Плюс – комбинацию клавиш для переключения раскладки можно выбрать на этом этапе. Минус – при определении разрешения экрана автоматика дала сбой, пришлось вмешиваться и указать правильное значение.

Существенное достоинство дистрибутива в том, что практически никакой постустановочной настройки не потребовалось. Мы лишь назначили более приятные глазу шрифты, но это уже дело вкуса – наверняка кому-то понравятся и те, что определены разработчиками.

Установка браузера *Chrome* закончилась неудачей. Версии специально для Альт Линукс на официальном сайте нет, а RPM-пакет, предназначенный для Fedora и openSUSE, оказался в данном случае бесполезен – система «жаловалась» на неудовлетворенные зависимости. Впрочем, можно установить свободный аналог этого приложения – браузер *Chromium*. Делается это при помощи штатного менеджера пакетов – программы *Synaptic*. Но и тут не обошлось без приключений. Дело в том, что при настройке беспроводной сети система не только запросила пароль доступа к маршрутизатору, но и попросила задать секретное слово для хранилища самих паролей. И если начать работу раньше появления окошка, в которое надо ввести тот самый суперпароль, система попросту виснет.

В принципе, логику разработчиков можно понять: торопыгам лучше держаться от компьютера подальше. Вот и воспитывают они у пользователя выдержку.

И еще один нюанс. Перед тем, как приступить к установке программ из сетевого репозитория, его надо активировать – по умол-

чанию система считает, что все программы хранятся на компакт-диске, с которого производилась установка.

А вот ПО для поддержки *DropBox* из репозитория устанавливать не стоит. Впрочем, этого сделать и не получится: несмотря на то, что пакет *nautilus-dropbox* входит в репозиторий, его установка закончится ничем. Впрочем, оно и понятно, ведь штатная графическая среда Альт Линукс – KDE, а *Nautilus* входит в состав GNOME.

Однако выход из ситуации может быть быстро найден на форуме Альт Линукс. Конечно, придется немного поработать в консоли, но набрать несколько команд – труд невеликий.

GPRS-модем Merlin XU870 распознал автоматически. Сразу после выбора страны и оператора связь была установлена. Вся процедура заняла буквально несколько секунд.

Кодеки для просмотра видео и прослушивания музыки в системе тоже имеются. Из сети ничего выкачивать не надо, что наверняка понравится тем, кто ограничен в использовании Интернета.

Поскольку этот дистрибутив разрабатывался специально для российских школ, то требования учебной программы в нем безусловно выполнены. Free Pascal установлен, а для удобства работы с ним имеются целых две оболочки – Lazarus и IDE, очень напоминающая Turbo Pascal как по внешнему виду, так и по функциональности (досадно только, что русские буквы в нем отображаются «кракозябрами»).

Получить бесплатную поддержку можно на форумах – как официальном, так и сообщества. Однако информация там не структурирована, поэтому ответ на вопрос придется поискать. Также этот дистрибутив не особо популярен у блогеров, так что кратких и внятных текстов по настройке системы Альт Линукс не так много. Зато сам разработчик выпускает учеб-

ники, которые пользователям этой системы безусловно нужно купить (или читать бесплатно в онлайн – доступ к текстам открыт).

К тому же, дистанционные курсы для учителей писались именно для пользователей дистрибутива Альт Линукс. Правда, предыдущей версии, но это не слишком существенно для новичка.

Платная техподдержка также оказывается – как самим разработчиком, компанией Альт Линукс, так и фирмой PingWin Software. Выбрать есть из чего.

Система Альт Линукс успешно справилась со всеми тестовыми задачами. Однако пользователю все же пришлось обращаться за помощью к сообществу – самостоятельно он бы вряд ли справился с задачей подключения *DropBox*.

«Требования учебной программы в Альт Линукс выполнены.»

Edumandriva 2010

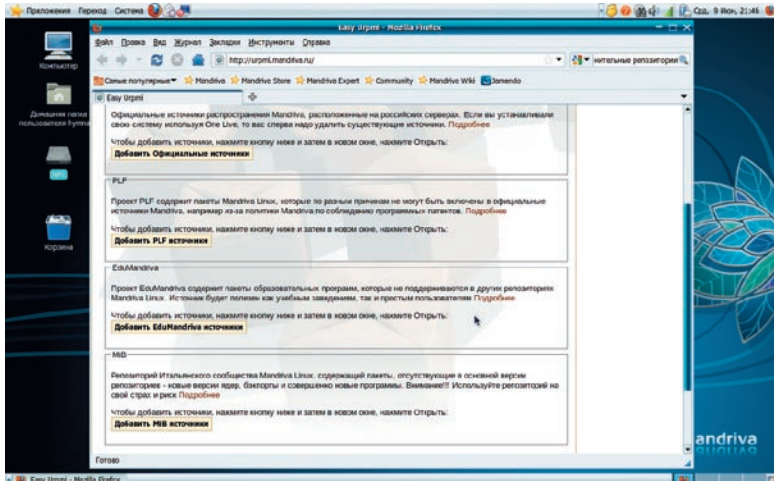
По всей видимости, разработчики Mandriva – большие поклонники Оккама. Они решили не плодить без нужды новые сущности. Так что Edumandriva – это не дистрибутив системы Linux в том смысле, который чаще всего вкладывается в этот термин.

Edumandriva – репозиторий, из которого можно установить ПО, необходимое для школы. Удобство такого подхода в том, что школа не привязывается к какой-то конкретной реализации системы. В качестве основы допускается как бесплатный Mandriva Free, так и коммерческий Powerpack. Дополнительная степень свободы, так сказать.

Получить дополнительное ПО можно различными способами – загрузить диск Edumandriva с сайта www.spohelp.ru или просто подключить дополнительный репозиторий на сайте urpmi.mandriva.ru. Результат должен быть одинаков.

Таким образом, рассматривать Edumandriva сам по себе – нелепо. Только в комплексе с базовым дистрибутивом – Mandriva. Именно так мы и поступим.

➤ Для подключения Edumandriva не обязательно скачивать диск.



Установка системы в автоматическом режиме не таит в себе никаких сюрпризов. Зная себе, щелчком мышки для продолжения процесса. На этой стадии определяются все основные параметры, включая комбинацию клавиш для переключения раскладок, поэтому пользователь получит практически готовое рабочее окружение сразу после завершения инсталляции.

А вот с GPRS-модемом произошла осечка. Система верно определила устройство и даже запросила PIN-код. Но потом попросила проверить, вставлена ли SIM-карта, которая была там, где ей и положено находиться. В общем, выходить за пределы действия Wi-Fi-сети пользователям Mandriva пока не рекомендуется.

Пакет *nautilus-dropbox* устанавливается штатно – из репозитория. Вы просто запускаете менеджер программ, вводите соответствующий поисковый запрос и даете команду на инсталляцию. Точно таким же способом устанавливается свободный аналог браузера производства Google – *Chromium*. Если же очень хочется «настоящий» *Chrome*, то и тут особых проблем нет – подойдет RPM-пакет с официального сайта.

К мультимедийным возможностям системы претензий нет. Если учителю потребуется показать детям учебный фильм, или ученику захочется дома послушать музыку, то даже дополнительных пакетов устанавливать не придется.

Free Pascal, а также оболочки Lazarus и Geany устанавливаются из репозитория Edumandriva. Если вы решили загрузить сразу диск, то в режиме инсталляции всего, что там находится, помимо них в системе появятся очень много приложений, предназначенных именно для школы.

Кстати, именно по этой причине предпочтительней выбирать загрузку всего диска, а не просто подключение репозитория. Если вы не можете похвастаться большим опытом в использовании Linux, то неплохо бы сперва познакомиться со всем ПО, чтобы выбрать то, которое нужно вам.

С технической поддержкой есть варианты. Первый – бесплатная помощь сообщества. Ресурсов, на которых ее можно получить, не так много, как у Ubuntu, но все равно достаточно.

Второй – фирменная поддержка компании Mandriva.ru. Третий – помощь фирмы PingWin Software. Опять же, есть из чего выбрать.

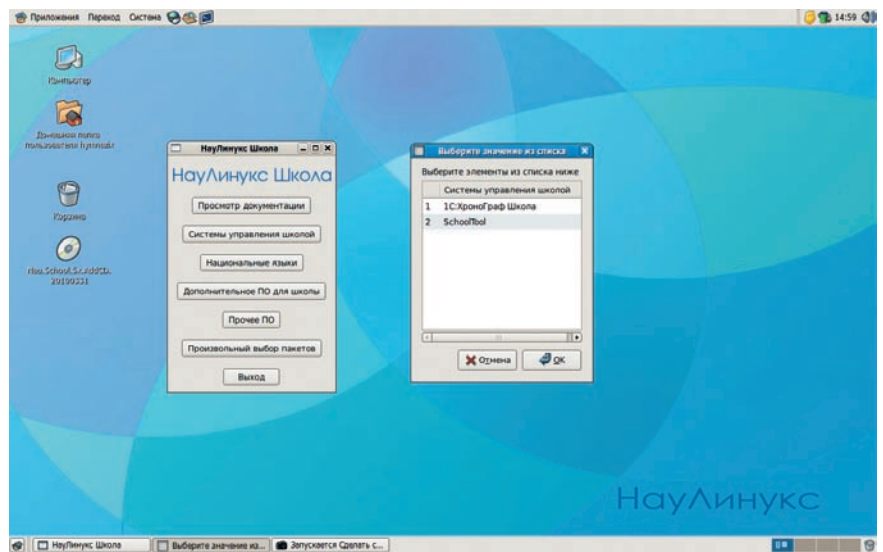
Система проста и понятна в эксплуатации. Однако с одной задачей она не справилась. Если вам не принципиальна поддержка сетей GPRS/EDGE, то ее вполне можно рекомендовать. Если же мобильность необходима, то следует проявить осторожность.

НауЛинукс 5.4

На сайте www.spohelp.ru размещены ссылки на несколько версий дистрибутива НауЛинукс. Это сделано вовсе не потому, что старые релизы имеют какие-то особенности, благодаря которым их можно предпочесть новым. Просто таким образом показывается, что техническая поддержка оказывается всей линейке продуктов. Если вы ранее не использовали НауЛинукс и вам не нужна какая-то конкретная версия, то лучше качать ту, что посвежее.

Автоматический режим установки предусмотрен и работает корректно. Первая заметная особенность инсталляции – требование к количеству знаков в пароле (не менее шести). Кому-то это понравится, кому-то – нет. С одной стороны, для школ это вроде плюс – длинный пароль гарантирует, что детишки не будут безобразничать в домашних каталогах друг друга (хотя это тоже спорный вопрос – вероятнее всего, они сами друг другу все расскажут). С другой – дома это откровенно неудобно.

Зато вторая наверняка понравится всем – удобный интерфейс для установки ПО с дополнительных дисков. Например, пользо-



ватель может загрузить с сайта www.spohelp.ru CD с программами для образовательных учреждений и установить любую из них буквально одним щелчком мыши. Разумеется, хоть на школьный, хоть на домашний компьютер – это важно для тех, кто работает еще и дома (а среди учителей таких людей наверняка больше половины).

Если по каким-либо причинам вам нужна мобильная рабочая станция, то НауЛинукс – дистрибутив не для вас. На подключение модема Merlin XU870 система отреагировала стоически. То есть – никак. Соответственно, о выходе в сеть не может быть и речи.

Кстати, не вполне адекватно ведет себя этот дистрибутив не только при попытке выйти в интернет по GPRS/EDGE/3G, но и через Wi-Fi. Дело в том, что штатный инструментальный настройки сети не признает существования WPA. Или Shared key, или Open system. Конечно, переконфигурировать маршрутизатор не сложно, но это явно лишняя работа. Вероятнее всего, дистрибутив изначально задумывался как средство развернуть рабочую станцию в проводной сети (в большинстве школ именно такая и есть). А о том, что учителя и ученики иногда пользуются домашней машиной в учебных целях, авторы позабыли.

Chrome (Chromium) и *DropBox* – результат тот же. В смысле, отрицательный. Причина невозможности установить пакеты, загруженные с официальных сайтов проектов, по большому счету понятны – дистрибутив НауЛинукс очень консервативен, в нем не используются самые свежие версии приложений.

Плюс это или минус – в общем случае сказать сложно. Для школы – наверное, плюс (старые проверенные программы работают, как правило, стабильнее). Для дома – вероятнее всего, минус (пользователь лишен возможности посмотреть на самые свежие и, не побоюсь этого слова, модные решения). Однако базовые развлекательные задачи (просмотр кино и прослушивание музыки) НауЛинукс решает – кодеки в системе имеются, да и проигрывателями система укомплектована с избытком.

Не будет проблем и у преподавателя информатики – Free Pascal и Lazarus устанавливаются из репозитория. Несмотря на то, что рассмотрение остального учебного ПО не входит в нашу задачу, нельзя не заметить, что НауЛинукс может подойти для первичного знакомства со школьным программным инструментарием – в состав системы входит очень много приложений, и все они корректно работают.

Отсюда и резюме. НауЛинукс – дистрибутив для школы и только для нее. В стенах учебного заведения он будет чувствовать себя превосходно. А вот выносить его за пределы этих стен, наверное, не стоит – для домашних машин учителям и ученикам лучше выбрать что-то другое. LXF

► В системе НауЛинукс очень удобно устанавливать дополнительное учебное ПО.

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ**Главный редактор**Валентин Сеницын info@linuxformat.ru**Литературный редактор**

Елена Толстякова

Переводчики

Илья Аввакумов, Александр Бикмеев, Юлия Дронова, Александр Казанцев, Ольга Кокорева, Светлана Кривошеина, Алексей Опарин, Валентин Развозжаев, Татьяна Цыганова

Редактор диска

Александр Кузьменков

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатели

Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Переплетная фабрика № 1»196210, Санкт-Петербург, Взлетная ул., 11, корп. 2, лит. А
Тел.: (812) 677-98-03
Заказ 3276**РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ****Редактор** Пол Хадсон [Paul Hudson] paul.hudson@futurenet.com**Редактор обзоров** Грэм Моррисон [Graham Morrison]graham.morrison@futurenet.com**Редактор диска** Майк Сондерс [Mike Saunders] mike.saunders@futurenet.com**Художественный редактор** Эфраин Эрнандес-Мендоса[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com**Литературный редактор** Эндрю Грегори [Andrew Gregory]agregory@futurenet.com**Подготовка материалов** Джоно Бэкон [Jonno Bacon], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Энди Ченнел [Andy Chappelle], Алекс Кокс [Alex Cox], Бен Мартин [Ben Martin], Шашанк Шарма [Shashank Sharma], Ник Вейч [Nick Veitch], Евгений Балдин, Сергей Голубев, Андрей Боровский, Дмитрий Дроздов, Семен Есилевский, Евгений Крестников, Александр Маджугин, Василий Олоничев, Андрей Сенов, Сергей Семенов, Михаил Смирнов, Алексей Федорчук
Художественные ассистенты Стейси Блек [Stacey Black], Джон МакЭлистер [John McAllister]**Иллюстрации** Крис Хедли [Cris Hedley], Крис Винн [Chris Winn], iStockPhoto**КОНТАКТНАЯ ИНФОРМАЦИЯ****UK:** Linux Format, 30 Monmouth Street, Bath BA1 2BWTel +44 01225 442244 Email: linuxformat@futurenet.com**РОССИЯ:****Санкт-Петербург (редакция):**

Лиговский пр., 50, корп. 15

Тел.: +7 (812) 309-06-86

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел/факс: +7 (499) 271-49-54

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru**Авторские права:** Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за 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.htmlЗа информацией о журнале, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>

© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



В сентябрьском номере

Firefox: битва за выживание?

Удастся ли одному из самых успешных открытых проектов в истории выстоять перед взрывным ростом популярности Google Chrome?

Доки

Панель задач и кнопка «Пуск» – тяжкое наследие Windows 95. Док же, напротив, инновационная функция Mac OS X. Узнайте, как воссоздать его в Linux.

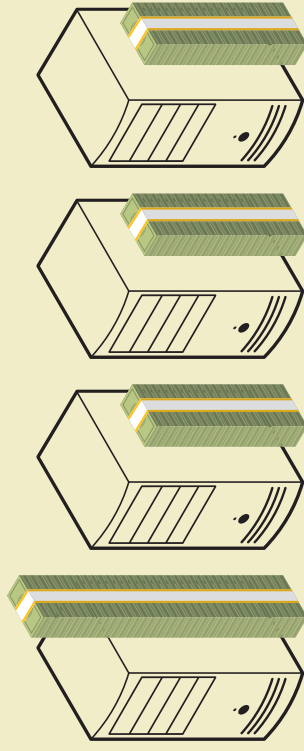
Hugin: советы и трюки

Как превратить хорошие отпускные снимки в прекрасные, используя свободное ПО и немного старания.

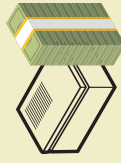
Содержание последующих выпусков может меняться без уведомления.

КОМПЛЕКСНОЕ РЕШЕНИЕ ДЛЯ МАЛОГО И СРЕДНЕГО БИЗНЕСА

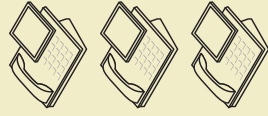
На базе Windows*



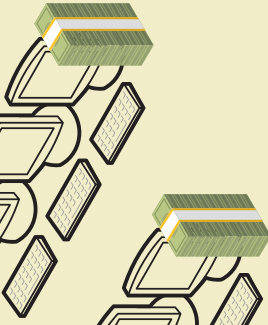
- MS WINDOWS SERVER**
Сервер приложений
IC:Предприятие
MS SQL
- MS WINDOWS SERVER**
Терминальный сервер
Почтовый сервер
MS Exchange
- MS WINDOWS SERVER**
Файловый сервер
- MS ACTIVE DIRECTORY**
Контроллер домена



Телефонная станция и IP-телефоны



Рабочие станции с Windows

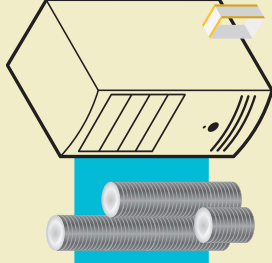


Терминальные рабочие станции и тонкие клиенты

На базе GNU/Linux и виртуализации*

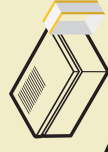
GNU/LINUX

- Терминальный сервер**
Клиент IC:Предприятие под Wine@Etersoft
IP-телефония Asterisk
Почтовый сервер Zimbra
Хранилище документов Altresko
- Терминальный сервер**
PostgreSQL
Сервер приложений IC:Предприятие
- Файловый сервер**
MDS Контроллер домена
DNS, OpenLDAP, WebConfig

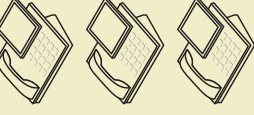


Внедрение и техническая поддержка силами специалистов GNU/Linuxцентра

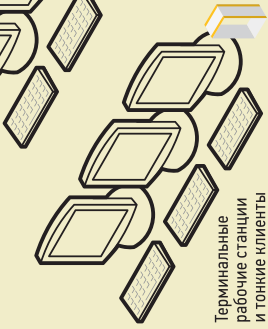
Почтовый сервер, файловый сервер, контроллер домена, базы данных SQL, IC:Предприятие, IP-телефония — и все это на одном физическом сервере в виртуальных машинах!



Телефонная станция и IP-телефоны



Рабочие станции с Linux



Терминальные рабочие станции и тонкие клиенты

ПРЕИМУЩЕСТВА ТЕРМИНАЛЬНОГО РЕШЕНИЯ НА БАЗЕ GNU/LINUX:

- экономия средств на лицензировании ПО
- сокращение затрат на оборудование за счет виртуализации
- единая инфраструктура на базе свободного ПО
- защита от вирусов и других угроз безопасности

Позвоните сейчас!

Назначьте встречу с нашим специалистом!

Офис в Санкт-Петербурге
Лиговский пр., 50, корпус 15
Тел.: (812) 309-06-86,
факс: (812) 640-49-90

Офис в Москве
Красноказарменная ул., 17
(в помещении АТС МЭИ)
Тел.: (499) 271-49-54



* Сравнительная стоимость программного обеспечения в комплексном решении для малого и среднего бизнеса.

Свободное ПО + виртуализация = экономия в 10 раз!

HOSTING NEXT LEVEL



Сэкономьте
340 рублей,
как новый
клиент!²

HETZNER
ONLINE
DEDICATED ROOT SERVER
**ЛУЧШЕЕ
ОБОРУДОВАНИЕ!
ЛУЧШИЙ СЕРВИС!
ЛУЧШИЕ ЦЕНЫ!**

HETZNER DEDICATED ROOT SERVER EQ 4

- Intel®Core™ i7-920 Quad-core с поддержкой технологии Hyper-Threading
- 8 GB DDR3 RAM
- 2×750 GB SATA-II HDD (Software-RAID 1)
- Операционная система Linux
- Windows Server от 550 руб. в месяц
- Неограниченный трафик¹
- Восстановление системы
- Установка из образов
- 100 GB пространства для резервных копий
- Без минимального контракта
- Стоимость установки 5900 рублей

1900
рублей в месяц

HETZNER DEDICATED ROOT SERVER EQ 8

- Intel®Core™ i7-920 Quad-core с поддержкой технологии Hyper-Threading
- 24 GB DDR3 RAM
- 2×1500 GB SATA-II HDD (Software-RAID 1)
- Операционная система Linux
- Windows Server от 550 руб. в месяц
- Неограниченный трафик¹
- Восстановление системы
- Установка из образов
- 100 GB пространства для резервных копий
- Без минимального контракта
- Стоимость установки 5900 рублей

3300
рублей в месяц

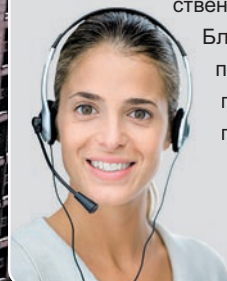
HETZNER DEDICATED ROOT SERVER EQ 9

- Intel®Core™ i7-975 Quad-core с поддержкой технологии Hyper-Threading
- 12 GB DDR3 RAM
- 3×1500 GB SATA-II HDD (Software-RAID 5)
- Операционная система Linux
- Windows Server от 550 руб. в месяц
- Неограниченный трафик¹
- Восстановление системы
- Установка из образов
- 100 GB пространства для резервных копий
- Без минимального контракта
- Стоимость установки 5900 рублей

3900
рублей в месяц

HETZNER ONLINE

Hosting Next Level (Хостинг нового уровня) означает, что компания Hetzner Online готова предоставить вам самые мощные решения для хостинга выделенных серверов из имеющихся сегодня на рынке. Наши предложения были разработаны, чтобы предоставить вам более высокую скорость и чрезвычайно стабильную сетевую инфраструктуру на базе наших собственных дата-центров в Германии. Благодаря лучшим ценам и непревзойденной поддержке, мы превосходим ожидания клиентов по всему миру.



www.hetzner.info
info@hetzner.com

¹ Трафик предоставляется бесплатно. При превышении порога 5000 GB/месяц скорость соединения будет ограничена 10 MBit/s. Стоимость аренды постоянного канала с пропускной способностью 100 MBit/s составляет 290 рублей за каждый дополнительный ТВ.

² Как новый клиент, вы можете сэкономить 340 рублей на первом платеже за любой из рекламируемых здесь продуктов. Просто используйте код ваучера 051108 при совершении заказа. Предложение действительно до 19 сентября 2010 года.