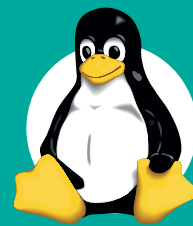


LINUX FORMAT

Главное в мире Linux

Ноябрь 2011 № 11 (150)



Прибыло ядро 3.0

Линус сказал свое слово.
Радуйся, мир свободного ПО! **с. 32**

ВАШ БЕСПЛАТНЫЙ DVD
Проба KDE 4.7 на Chakra Linux,
CentOS, Ubuntu 11.10a3 и более того!

ДИСТРИБУТИВ СВОИМИ РУКАМИ

Сделав собственную версию Ubuntu, SUSE
и Fedora, будем править миром дистрибутивов

ПЛЮС!
Сказке
о Linux
далеко не конец
с. 44

KDE 4.7 стал правильным

» Пользователи Gnome 3, вот как выглядит
достойный рабочий стол **с. 10**



Scientific Linux

» Если он хорош для CERN, то хорош
и для нас **с. 12**



Роб Пайк

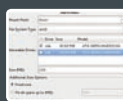
“ Google Go задуман
для тех вещей,
которые делает Google ”

И это название дано не зря **с. 30**

Также в номере...

Mageia без маски

» Чего добились ребята из Mandriva **с. 36**

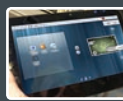


LVM надо любить

» Держите жесткий диск
под контролем с Fedora **с. 68**

Шипы гравитации

» Блестящая новая инди-игра хочет
в лицо дурацким теориям Ньютона **с. 13**



Настольный саммит

» Совместный мозговой штурм
разработчиков Gnome и KDE **с. 40**

ChorusOS

» Линуса на нее не было... но как знать? **с. 54**



Ключевые навыки

Защита данных

» Заслоните свою машину
от сетевых нападок

Управление контентом

Создаем wiki

» Потрясите умы
с сервером MediaWiki

Разделение оборудования

Синергия машин

» Мышь и клавиатура одна,
компьютеров много

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

Linux center
www.linuxcenter.ru

Red Hat Enterprise Linux

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

Самая популярная в мире
Linux платформа для бизнеса

Обеспечивает высокую
производительность, надежность,
масштабируемость и безопасность

Сертифицирована ведущими
производителями оборудования
и разработчиками ПО



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

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

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



«ГНУ/Линуксцентр» — Linux-эксперт для вашего бизнеса

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

Red Hat — ведущий серверный дистрибутив Linux

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

Специальное предложение для читателей Linux Format!

Закажите Red Hat Enterprise Linux в «ГНУ/Линуксцентре» до 30 мая
и получите в подарок книгу «Полное руководство пользователя Red Hat Enterprise Linux»



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

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 текущем месяце наша бригада-ух понаделала собственных дистрибутивов. Мы спросили их: чем эти дистрибутивы замечательны?



Майк Сондерс
В моем работает *PulseAudio!*. Минуточку, это вы всерьез или хотели, чтоб я пошутил?



Эндрю Грегори
У моего на обоях мой портрет. Всем прославлять меня! По любому, угостите пивком.



Эфраин Эрнандес-Мендоса
Мой дистрибутив взял лучшее от 32- и 64-битных дистрибутивов и работает в режиме 48 бит.



Нейл Ботвик
Все, кто работает в моем дистрибутиве, получают киндер-сюрприз. Если он есть в репозитории.



Маянк Шарма
А мой содержит только ядро и *wget*, а при наличии Интернета грузит новый дистрибутив стохастически.



Джонатан Робертс
В рекламе моего упомянуты «web 3.0», «облачные инициативы» и «бункеры данных». Стало быть, он лучший.



Энди Ченел
Мой дистрибутив применяется в IBM – один тип оттуда, тесть двоюродного брата моего приятеля, заглянул на его сайт!



Валентин Синецын
Мой работает на терере и может варить кофе – хороший выбор для кухни.



Ник Вейч
Я лицензировал песню «Не подходи ко мне больше никогда» как вводную музыку. Ну что, я победил?



Сюзан Линтон
Мой дистрибутив запускает себя из *VirtualBox*, затем в *VirtualBox* внутри себя, затем *explodes*.



Шашанк Шарма
Название моего – четырехмерный рекурсивный акроним, который только я и могу озвучить. Круто, да?



Боб Мосс
Мой дистрибутив пресекает войнушки, при загрузке выбирая рабочий стол и текстовый редактор случайным образом.



Сделай это сам

» Тема номера – создание собственных дистрибутивов – на мой взгляд, полностью соответствует основополагающей идее всего свободного ПО – DIY (Do-It-Yourself – сделай это сам). Если хочешь, чтобы что-то было сделано хорошо – сделай это сам... Именно так рассуждали все великие инноваторы XX века (от Томаса Эдисона до Линуса Торвальдса), приступая к реализации очередной неочевидной идеи. По крайней мере, Linux появился на свет именно так.

Из-за недостатка места в статье не рассмотрена «высшая и последняя стадия» самостоятельного создания собственного дистрибутива – Linux From Scratch (<http://www.linuxfromscratch.org>), что можно перевести на русский как «Linux с нуля». В рамках этого проекта разрабатывается методика (изложенная в одноименной книге) сборки работающей системы из исходных кодов. В ряде случаев – например, для портирования Linux на нестандартную аппаратуру – это единственно возможный путь. В одном из будущих номеров журнала постараемся написать об этом подробнее.

Идея DIY все шире распространяется по планете. Можно найти руководства по самостоятельному созданию практически чего угодно (от детских игрушек до легких самолетов включительно). А уж создать свой дистрибутив... это точно стоит попробовать.

Кирилл Степанов

Главный редактор

» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

Web-сайт: www.linuxformat.ru

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

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

Содержание

Всмотрись в журнал, читатель, и нырни в проекты этого месяца.

Обзоры

KDE 4.7 10
Свежий релиз среды рабочего стола.



» Куча всякого добра для KDE.

Scientific Linux 12
Клон RHEL, который явно намеревается задать жару CentOS.

VVVVVV 13
Рулите силами тяготения, чтобы ваш космический экипаж прорвался через этот ретро-платформер.



» Угловая графика и сигналы «бип-бип».

Digicam SC 2.0 14
Лучший менеджер фото под Linux опять похорошел.

Midori 0.4.0 15
Зелененький он был... это по-японски.

**Сравнение:
Web-браузеры** с. 16



ДИСТРИБУТИВ СВОИМИ РУКАМИ

Создайте свой крутой дистрибутив с. 22



Даешь электронную документацию! с. 48



Что за штука — OAuth? с. 56

Люди говорят



Мы подумали:
а не создать ли язык
программирования
с быстрой сборкой?

Роб Пайк — про свой интерес к Google Go с. 30

На вашем бесплатном DVD



CentOS 6

- » Тест-драйв KDE 4.7 в Chakra Linux
- » Образчик Ubuntu 11.10
- » Ретро-игры и инструменты разработчика

ПЛЮС: Подкасты, код, учебники... **с. 100**

Ищите в этом номере



Ядро 3.0 32

Не в номере счастье. Вскроем упаковку.

Внутри Mageia 36

Мистерия магии завораживает...

Desktop Summit 2011 40

Представители рабочих столов-лидеров обсуждают дорогу в будущее.



Пропустили номер?

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



Учебники

- Начинающим**
Мастер-класс по обороне 64
Защитите свою систему от онлайн-угроз.
- Управление памятью**
LVM 68
Разбиения диска не надо бояться!
- Wiki в Web**
MediaWiki 72
Назло Википедии, сделаем свою такую же.
- Командная строка**
Маленькие хитрости 76
Скрипт выписывает слова в тетрадку.
- Языки программирования**
Erlang 80
Базовые сущности исчерпаны полностью.
- Hardcore**
Synergy 84
Слаженная работа всех ваших машин.

Постоянные рубрики

- Новости** 4
Конкурс по НПП завершен, Linux садится на Ford, электроника рвется в облака, а Microsoft оттяпывает очередную кус.
- Сравнение** 16
Пять лучших web-браузеров препарированы и выстроены по ранжиру.
- Интервью LXF** 30
Роб Пайк симпатизирует сусликам и не любит медленной компиляции.
- Что за штука** 56
OAuth: вроде гардеробчика, но не для пальто, а для сайтов и данных.
- Рубрика сисадмина** 58
Восемь ступенек в сисадминский рай, плюс как уследить за временем с начала эпохи.
- Ответы** 88
ПРОБЛЕМЫ LINUX РЕШЕНЫ!
Хандрит *NoteHub*? Проблемы с *Plymouth*? Помощь рядом.
- Hotpicks** 94
Отведайте горяченького: лучшие в мире новинки свободного ПО.
- Диск Linux Format** 100
Содержимое двустороннего DVD этого месяца.
- Пропустили номер? ...** 104
Еще не поздно заказать любой из предыдущих выпусков журнала.
- Школа LXF** 108
Не дадим ученикам запугать учителей вопросами – возьмем да и ответим на них заранее.
- Через месяц** 112
Мы возвещаем появление на горизонте чудесного мира Linux.



» Объедините свои компьютеры.



ГЛАВНОЕ Arduino для ARM » Ford в социальных сетях » Прогресс НПП
» Casio платит Microsoft » Умер один из отцов Unix

КОНКУРС ЗАВЕРШЕН

ТП НПП – отсчет пошел!



» Рубрику готовил
ТИМУР МУБАРАКШИН

НПП переходит в фазу активного развития.

26 сентября 2011 года были опубликованы результаты конкурса, проводившегося Министерством связи и массовых коммуникаций Российской Федерации. Лот № 4 – «Разработка прототипов базовых программно-технических компонентов Национальной программной платформы и документов, регламентирующих порядок сборки, приемки, размещения и эксплуатации программных решений в фонде алгоритмов и программ (ИО/04-11)» – признан состоявшимся: выиграла российская компания «ПингВин Софтвар», предложив минимальную цену – 5 млн рублей. Через 16 дней после заключения госконтракта она должна представить прототип НПП.

По российским законам, контракт будет заключен не ранее 10 и не позднее 20 дней после объявления результатов. По его подписании сразу публикуются план-график работ и материалы по ним. Компании-«соучастники» работ – ВНИИНС, НЦПР, ГНУ/ЛинуксЦентр, «Линукс Инк» и питерский Политех; полный их список также вывесят на сайте проекта.

В открытом письме на сайте некоммерческого партнерства РАСПО <http://raspo.ru/> «ПингВин Софтвар» отмечает, что компания была одним из подписантов письма главе Совета по развитию информационного общества при президенте Сергее Нарышкину с просьбой отменить конкурс из-за ряда положений, не соответствующих законодательству. Также в письме говорится, что «ПингВин Софтвар» победил потому, что собрал самую большую команду – 147 сотрудников, в чем им помогли другие участники РАСПО, предоставившие свои ресурсы, и в данный момент задействовано уже 215 человек: к победителям присоединился еще один участник конкурса – ВНИИНС им. В. В. Соломатина. В итоге «ПингВин Софтвар» предполагает

передать заказчику несколько вариантов ОС, на базе разных дистрибутивов Linux. Помимо ОС, прототип НПП включает:

- » прототип эталонной среды разработки, сборки и обновления ОС и прикладных приложений на основе свободного ПО;
- » прототип эталонной ОС, включая пакет общесистемного ПО, и прототип ПО управления базами данных, на основе свободного ПО, с учетом требований по информационной безопасности;
- » прототип системы публичного доступа к общесистемным и прикладным компонентам НПП из фонда алгоритмов и программ.

«Нужно учиться зарабатывать, а не ждать госфинансирования.»

«ПингВин Софтвар» заявляет, что заказчик (Минкомсвязь) получит результаты вовремя. С их точки зрения, наполнение НПП должно происходить на конкурентной основе, а создаваемые в рамках НПП решения – выкладываться в репозиторий в Интернете, доступный всем желающим.

Политикой государства в отношении развития НПП, однако, довольны не все заинтересованные участники. Павел Фролов, генеральный директор компании «Мезон» (ГНУ/ЛинуксЦентр), говорит, что идея НПП правильная, но шаги, предпринятые сейчас государством, неверны. Он считает, что нужно действовать по-другому: финансировать создание решений для НПП не через открытый конкурс, а в форме грантов, избегая монополизации любого из направлений НПП. Также, по его мнению, все ПО, разработанное на деньги государства (то есть народа), должно публиковаться в фонде алгоритмов и программ под сво-

бодной лицензией как типовые проектные решения. Проприетарные же решения могут публиковаться в этом фонде, если разработчик предоставит их для бесплатного применения госорганизациями.

Дмитрий Комиссаров, глава «ПингВин Софтвар», опасается, что концерн «Сирис», координатор НПП, будет ущемлять конкурентов: без публичных обсуждений и контроля со стороны общественных организаций ничто не помешает возврату к идеям единственного вендора и координатора, принимающего решения. По его мнению, для достижения успеха нужно учиться зарабатывать на рынке СПО, а не дожидаться госфинансирования.

Алексей Новодворский, зам. генерального директора компании «АЛТ Линукс», тоже участника конкурса Минкомсвязи, заявил в интервью изданию «Компьютерра», что «Алты Линукс» вышла из НП «РАСПО», не желая ассоциироваться с решениями, деструктивными для развития НПП (речь идет о письме с просьбой об отмене конкурса), и не имея возможности на них влиять. В понимании «Алты Линукс», НП «РАСПО» стало бизнес-организацией, а это расходится с целями, согласованными при создании РАСПО.

Развенчание мифа

Гуляющий по Сети миф о том, что НПП – это теперь непременно и окончательно Mandriva (РОСА), опровергнут. По словам генерального директора ОАО «Линукс Инк» Леонида Сомса, методология работы команды РАСПО при выполнении проекта обеспечит такой результат, что любой производитель программных продуктов, удовлетворяющих разработанному набору стандартов и регламентов, будет иметь возможность участвовать в любых поставках, а эти продукты – быть включенными в фонд алгоритмов и программ. Итак, в НПП найдется место достойным решениям на основе СПО. Также эксперты сходятся во мнении, что заявленная сумма контракта в 5 млн рублей не покроет даже издержек на разработку, и уж тем более бессмысленно говорить об «откатах» и «распилах».

ПОЕХАЛИ

Ford и Linux – вместе навсегда?

Ford создает платформу для разработки открытых дополнений к автомобильной информационной системе.

Компания Ford объявила недавно о работе над новой платформой OpenXC для автомобильных информационно-развлекательных систем, которая базируется на идеях взаимозаменяемости программных и аппаратных компонентов, тесном использовании интернет-служб и задействовании технологий, применяемых в современных смартфонах. Платформа использует аппаратные компоненты BugBases и BugModules, а также облачный сервис BugSwarm. OpenXC предоставляет возможности разработки открытых приложений и модулей для автомобилей Ford – дополнений для работы со звуком и системами безопасности, а также подключения внешних сенсоров, например, датчика дождя.

Ford работает совместно со стартапом Bug Labs, разработавшим открытую платформу Bug System, которая позволяет комбинировать отдельные программы и аппаратные модули в зависимости от конкретных нужд владельца. Программной начинкой Bug System служит Linux, который работает на платформе OMAP3530

с процессором ARM Cortex 600 Мгц. А облачная служба BUGswarm обрабатывает и предоставляет доступ к данным через глобальную сеть. В качестве примера приводится вариант приложения, которое позволит организовать через социальные сети обмен информацией об экономии топлива в привязке к местоположению.

О начале сотрудничества двух компаний было объявлено на конференции TechCrunch Disrupt в Сан-Франциско. Заявлено, что платформа OpenXC позволит быстро создавать прототипы автомобильных информационно-развлекательных систем и проводить их тестирование. На международной выставке Motor Show IAA во Франкфурте Ford собирается представить свой новый концепт-кар Evos – машину будущего, подключенную к социальным сетям.

В течение 2012 года Ford планирует экспериментальное распространение в Европе информационно-развлекательной системы SYNC, которая включает голосовое управление, систему навигации и развлекательные службы. SYNC как раз и основана на открытой платфор-



➤ Автомобиль Ford Evos станет частью интернета, но нужно ли это людям? Вопрос пока открыт.

ме OpenXC, предназначенной для расширения функциональности автомобильных информационно-развлекательных систем. В будущем, по словам разработчиков, автомобиль превратится в док-станцию для взаимозаменяемой аппаратуры plug-and-play, созданной компанией Bug Labs, и программных модулей, содержащих только требуемые хозяину машины коммуникационные функции и службы.

ЭЛЕКТРОНИКА В ОБЛАКАХ

Микроконтроллеры Wedana

Облачное проектирование электронных схем стало реальностью.

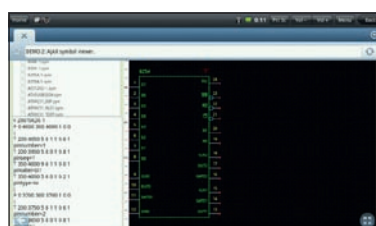
В августе этого года завершился очередной этап разработки проекта Wedana, в рамках которого создается система для проектирования электронных схем, работающая через обычный web-браузер. Проект создан с целью обеспечить возможность редактирования электронных схем на платформах, на которых не могут выполняться существующие

системы проектирования: например, на мобильных системах типа планшетов на базе Apple iOS и Google Android. Кроме того, использование web-технологий позволит подготовить среду для совместного проектирования схем, чтобы над одним проектом могли одновременно работать несколько инженеров.

На текущем этапе развития проекта реализована возможность просмотра файлов в формате gschem, подготовленных в свободном редакторе принципиальных схем gEDA.

Также подготовлен модуль для интеграции с системой совместной разработки Trac. Код проекта написан на языке JavaScript и распространяется под лицензией GPL. При разработке Wedana используются стандартные web-технологии, поддерживаемые свежими версиями

популярных web-браузеров, такие как тэг Canvas. Несколько онлайн-демонстраций работы Wedana можно посмотреть на сайте <http://sourceforge.net/apps/trac/wedana/wiki/Demo>.



➤ Благодаря проекту Wedana проектирование схем стало проще.

Дружим домами

В последние годы в мире открытого железа вошло в моду сотрудничество мелких групп независимых разработчиков с крупными компаниями, такими как Ford или Intel.

Это хорошая тенденция, и она не случайна. Принцип открытой разработки чего бы то ни было привел к появлению серьезных конкурентов проприетарным решениям. Примером может послужить сообщество Mozilla и их популярные во всем мире программы. По мнению некоторых экспертов, в будущем будет стерта грань между открытыми и закрытыми разработками, и появится новая концепция, позволяющая создавать полностью или почти открытые спецификации и устройства, стоимость которых будет значительно снижена по сравнению с ныне существующими.

ОТКРЫТОЕ ЖЕЛЕЗО

ARM на Arduino

Проект Arduino вступил в новую фазу своего развития.

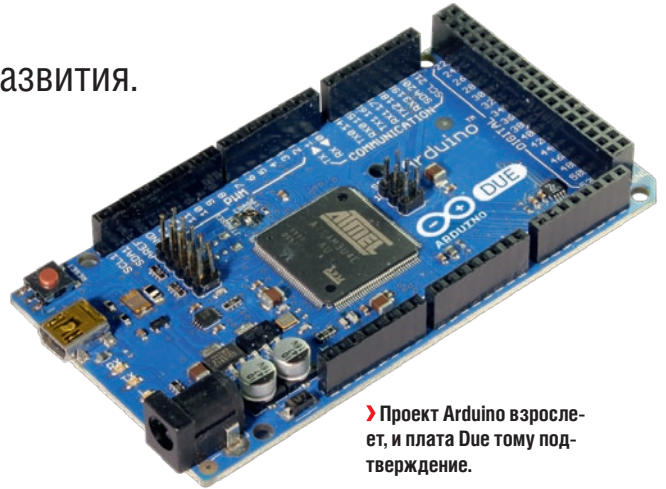
В рамках проекта Arduino представлена новая плата – Arduino Due. Продукт примечателен тем, что использует процессор на базе архитектуры ARM. Предыдущие платы Arduino изготавливались на основе чипов ATmega американской компании Atmel. Как и прочие разработки Arduino, плата Arduino Due имеет полностью открытые спецификации под лицензией Creative Commons BY-SA. Любуй интересующийся может скачать схемы платы, внести улучшения и изменения в них, и даже начать производство своих продуктов.

Основа Arduino Due – 32-разрядный процессор Atmel SAM3U, работающий на частоте 96 МГц, построенный на базе ARM-ядра Cortex-M3 и соответствующий архитектуре ARMv7-M. Arduino Due имеет на борту 256 КБ Flash-памяти и 50 КБ SRAM, пять SPI-шин (Serial Peripheral Interface bus – последовательный периферийный интерфейс), два I2C-интерфейса (для работы со звуком), пять портов UART (для связи с цифровыми устройствами) и 16 аналоговых входов. Для подключения

периферийных устройств на плате имеется порт USB.

Работа на Arduino Due еще не закончена. Распространение финальной версии разработчики планируют начать в конце 2011 года. В данный момент всем заинтересовавшимся платформой людям предлагается принять участие в ее разработке, заказав через сайт store.arduino.cc предварительный демонстрационный образец.

Разработчики проекта Arduino также объявили о выпуске упрощенного варианта классической платы Arduino Uno – Arduino Leonardo. Leonardo имеет ту же форму и те же разъемы, что и Uno, но отличается более упрощенной схемой на основе контроллера Atmega32u4. Для Arduino Leonardo уже готов драйвер USB, симулирующий мышь, клавиатуру и последовательный порт. Для программирования микроконтроллера разработан свободный SDK под лицензией GPLv2, который поддерживает разработку на специально созданном языке программирования. Также на днях стал доступен для загрузки финальный релиз Arduino API 1.0.



► Проект Arduino возрел, и плата Due тому подтверждение.

В дополнение ко всему, разрабатывается модуль Arduino Wifi Shield, предназначенный для обеспечения устройств Arduino средствами беспроводной связи. Основанный на микро-wifi чипе компании H&D Wireless и дополненный процессором AVR32, позволившим реализовать поддержку полноценного TCP/IP стека внутри модуля, Arduino Wifi Shield призван обеспечивать максимальные возможности по своей модификации. Пользователи могут реализовать свои протоколы и адаптировать устройства под свои нужды.

НАРУШЕНИЯ ПО СЕКРЕТУ

Microsoft нейдет

Microsoft продолжает забирать якобы свое от производителей оборудования.

Мicrosoft в очередной раз создала важный прецедент, способный негативно повлиять на дальнейшее развитие всей экосистемы Linux. Компания Cisco заключила с софтверным гигантом соглашение, в котором признается наличие в коде Linux технологий, нарушающих патенты Microsoft, и приняла на себя обязательство выплачивать лицензионные отчисления за использование Linux в своих продуктах. Размер отчислений не сообщается, условия сделки объявлены конфиденциальными. Список нарушенных патентов Microsoft также не обнародован.

За последние месяцы Microsoft заключила несколько сделок с крупными и средними компаниями, обязующих эти компании выплачивать лицензионные отчисления за использование платформы Android. Сделки были заключены с Acer,

ViewSonic, HTC, Vilocity Micro, General Dynamics, Onkyo и Wistorn Corp. Компании Barnes and Noble и Samsung отказались заключать соглашение с Microsoft, объявив требования абсурдными, а стоимость лицензирования непомерно высокой. Если ранее Microsoft пыталась всеми силами снизить привлекательность Android в глазах производителей мобильных устройств и добиться повышения цены для конечных потребителей, то теперь усилия сосредоточились на дискредитации любых решений на базе Linux.

Раньше Microsoft уже не раз высказывала спекулятивные заявления, что в Linux якобы нарушаются ее патенты. Однако дальше угроз и заключения формальных соглашений о взаимном обмене интеллектуальной собственностью дело не доходило. Сейчас же Microsoft переступила черту, нарушила и без того шаткий «статус кво»

и неприкрыто начала сбор денег за использование Linux.

В 2007 году компания Microsoft заявляла о нарушении в Linux и открытом ПО 235 патентов: Linux-ядро – 42 патента, элементы графических интерфейсов пользователя (меню, панели инструментов) – 65 патентов, пакет *OpenOffice.org* – 45 патентов, почтовые клиенты – 15 патентов, другие открытые программы – 68 патентов. Указанные патенты так и не были раскрыты, а все связанные с ними сделки проводились с обязательной подпиской о неразглашении. Благодаря этим патентам, Microsoft удалось заключить патентные соглашения с такими компаниями, как Novell, Fuji-Xerox, Samsung и Xandros. Компании Mandriva, Canonical и Red Hat отказались подписывать соглашение с Microsoft, и никаких судебных разбирательств после этого не последовало.

ВЕЧНАЯ ПАМЯТЬ

Не стало Денниса Ричи

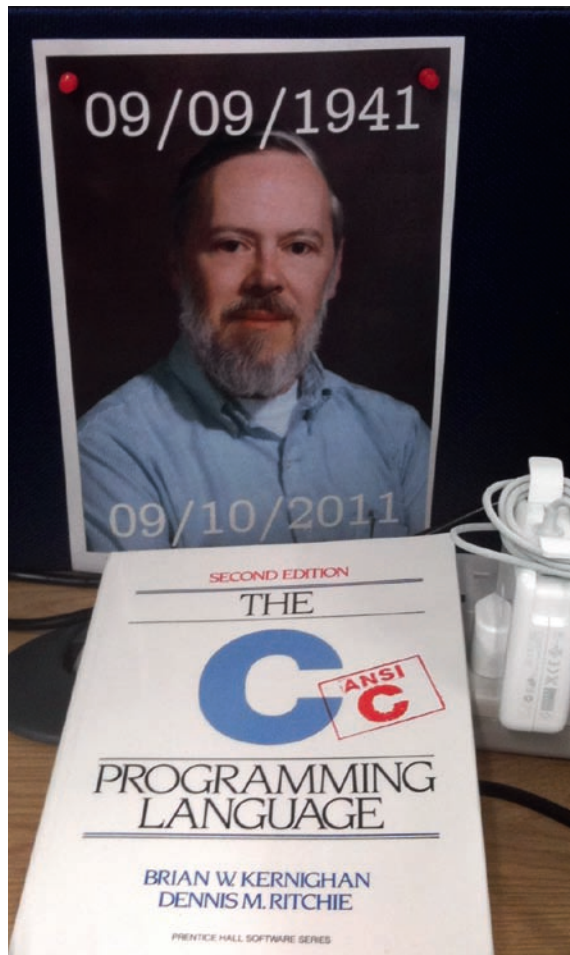
Ушел из жизни один из создателей Unix и языка Си.

Октябрь стал черным месяцем для индустрии ИТ. Вслед за смертью Стива Джобса [Steve Jobs] пришла еще одна скорбная новость. Роб Пайк, один из создателей оригинальной Unix, сообщил о смерти Денниса Ричи [Dennis MacAlistair Ritchie], создателя языка Си и одного из ключевых разработчиков операционной системы UNIX. Впоследствии Ричи принимал участие в разработке ОС Inferno и Plan 9, а также языка Limbo. Ричи умер 8 октября у себя дома в городе Мюррей-Хилл, в возрасте 70 лет, после тяжелой продолжительной болезни. Причиной смерти стал рак простаты.

Ричи родился в Бронксвилле, Нью-Йорк, окончил Гарвард со степенью бакалавра по физике и прикладной математике. В 1967 году начал работать в Bell Labs.

Трудно переоценить влияние этого человека на развитие ИТ-индустрии, как, впрочем, и на весь технический прогресс в последние 40 лет. Созданные Ричи идеи, философия и технологии существенно повлияли на развитие всей компьютерной индустрии и отражаются практически в любом программном продукте. В 1983 году он был награжден премией Тьюринга, в 1990 году – отмечен медалью Ричарда Хэмминга от Института инженеров электротехники и электроники (IEEE), в 1999 году получил из рук президента США Билла Клинтона Национальную медаль за достижения в области технологий. В 2011 году он был награжден Призом Японии за достижения в области информации и связи.

Выпущенная в соавторстве с Брайаном Керниганом книга «Язык программирования Си» стала отправной точкой для миллионов разработчиков. Русскоязычное издание данной книги было одним из первых руководств



» По этой замечательной книге выучилось не одно поколение программистов.

по языку Си, выпущенных массовым тиражом (первое издание вышло в 1985 году, а второе, с тиражом 100 тыс. экземпляров, в 1992 году).

«Влияние Ричи на развитие ИТ-индустрии трудно переоценить.»

Время великих

Достижения хакеров второй половины XX века до сих пор являются ключевыми для всей индустрии ИТ. Дух тех великих времен был духом истинного хакерства и изобретательства. Именно такие пионеры, как Деннис Ричи, Стив Джобс или Дуглас Энгельбарт [Douglas Engelbart], человек, изобретший знакомую

всем мышь, а также всю концепцию графического интерфейса пользователя, заложили фундамент для развития современных «высоких технологий». А мы до сих пор пользуемся графическим интерфейсом, указывающим устройством и, конечно, персональными компьютерами.

Новости короткой строкой

» К 2012 году в госучреждениях Парагвая планируют использовать только открытое ПО – Linux, Libre Office и прочее.

Источник: www.osor.eu

» 14 октября 2011 года проекту KDE исполнилось 15 лет. Первый стабильный релиз KDE 1.0 был выпущен 12 июля 1998 года, после почти двух лет активной разработки.

Источник: <http://kde.org>

» 13 октября 2011 года было подписано соглашение о практическом использовании свободного программного обеспечения в секторе государственного управления на территории Самарской области.

Источник: <http://tp-npp.ru>

» Увидел свет релиз дистрибутива Ubuntu 11.10 «Oneiric Ocelot». Дистрибутив доступен в редакциях для настольных компьютеров, серверов и облачных окружений.

Источник: <https://lists.ubuntu.com>

» В списке рассылки пользователей Wine объявлено о получении доступа злоумышленников к СУБД, используемой для обеспечения работы каталога приложений (appdb) и системы отслеживания ошибок в проекте Wine, что привело к утечке базы паролей пользователей Bugzilla и AppDB.

Источник: www.winehq.org

» Российская компания РусБИТех, развивающая дистрибутив Astra Linux, вступила в Linux Foundation. Это третья российская компания, подключившаяся к работе Linux Foundation; ранее в данную организацию вошли компании Parallels и PingWin Software.

Источник: www.marketwatch.com

» Фонд свободного ПО объявил о признании проекта GNU PDF состоявшимся и исключении его из списка приоритетных проектов. Целью проекта GNU PDF было создание и распространение свободного набора высококачественных и многоплатформенных компонентов, позволяющих работать с форматом PDF в полном соответствии со спецификацией PDF 1.7 и стандартом ISO 32000-1.

Источник: www.fsf.org **1X3F**

ПОДПИШИСЬ НА ЖУРНАЛ LINUX FORMAT!

2012

LINUX
FORMAT
Главное в мире Linux



Оформи в редакции
подписку на печатную
версию журнала
и получи в подарок
диск с архивом номеров,
а также подписку
на электронную версию
издания в формате PDF.



Стоимость подписки Годовая — 2160 руб., полугодовая — 1170 руб. без учета стоимости доставки.
Адреса и телефоны редакции Санкт-Петербург, Лиговский пр., 50, корп. 15, тел. (812) 309-06-86.
Москва, Красноказарменная ул., 17, тел. (499) 271-49-54.

Варианты доставки Почтой по России простой бандеролью — журнал доставляют прямо в почтовый ящик
» Почтой по России заказной бандеролью — в почтовый ящик приходит извещение, номера выдают на почте » Курьером
«ГНУ/Линуксцентра» по Москве и Санкт-Петербургу » Курьерской службой СПСР по России » В виде PDF-файлов
для подписчиков электронной версии.

shop.linuxformat.ru



Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
...простой советский юзер.

Дети мага Мандрейка

Драмматическая история Mandriva (в 1998 году, под именем Mandrake, он стал первым настоящим пользовательским дистрибутивом) закончилась год назад расщеплением на две независимые линии: Mageia, ведомый прежней командой, и собственно Mandriva, уже представляющий в значительной мере российскую разработку.

Вышедший летом релиз Mageia представлял собой косметическую доработку исходной системы. А вот недавний релиз Mandriva 2011 уже содержит радикальные новшества — начиная от изменения формата пакетов на rpm5 и внедрения systemd и заканчивая основательной переделкой рабочего стола KDE. В результате система получилась фронтальная, очень эстетичная, но весьма тяжеловесная — даже по сравнению с Mageia, тоже не эталоном легкости.

И это повод вспомнить о более раннем ответвлении Mandriva — PCLinuxOS, давно уже превратившемся в самостоятельный дистрибутив с собственной концепцией и пакетной базой. Она не очень обширна, но зато система оказывается очень компактной и аккуратной. До недавнего времени она существовала только в 32-битном варианте, но в настоящее время полным ходом идет тестирование 64-разрядной версии, релиза которой можно ожидать в ближайшее время.

Таким образом, старым поклонникам мага Мандрейка предоставляется достаточно широкий выбор.

alv@posix.ru

Сегодня мы рассматриваем:

- KDE 4.7** 10
Спустя годы после скандального KDE 4.0 вышел KDE 4.7 — и, кажется, лучшего и желать нельзя: он элегантен, щедр на функции и выглядит классно. Но залижет ли он наши раны от 4.0? А точнее, оттянет ли он людей от Gnome 3?
- Scientific Linux 6.1** 12
Берем кодовую базу Red Hat, откидываем все небесплатные части и встраиваем поиск бозона Хиггса — получается Scientific Linux. И частице по имени Бог он понравился бы, и простые смертные останутся довольны.
- VVVVVV** 13
Вверх-вниз и вверх тормашками — в этой блистательной игре-платформере вы вволю наощущаетесь головокружения. Но нельзя же не выручить своих соратников из ловушек! Иначе ваш космолет останется без экипажа...
- Digicam 2.0** 14
Умники любят свободное ПО и любят цифровые камеры — неспроста в Linux такое количество отличных менеджеров фотографий. Перед вами — новый основной релиз одного из лучших представителей данной категории.
- Midori 0.4.0** 15
Быстрый и нетребовательный к ресурсам web-браузер — жаль только, не располагает документацией на русском языке. Но уж по-английски все объяснения довольно подробные и обстоятельные.



» Благодаря своему удобству, KDE станет тихой гаванью для пользователей, раздосадованных Gnome и Ubuntu.



» Лютая смерть на клавишах подстерегает зазевавшихся в VVVVVV — этом гимне бренности всего живущего.

Наш вердикт: Пояснение

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



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

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

LINUX FORMAT Вердикт

Braid

Разработчик: Number None
Сайт: www.braid-game.com
Цена: \$9,99 (ок. £6,15)

Игровой процесс	10/10
Графика	10/10
Простота использования	9/10
Оправданность цены	10/10

» Braid сумеет увлечь своим интригующим шармом даже не охочих до платформеров и головоломок.

Рейтинг 10/10



KDE 4.7

4.7 доверху набит новинками, но отмоет ли он запятнанную репутацию рабочего стола? **Шашанк Шарма** разбирается.

Вкратце

» Одно из двух главных рабочих окружений настольного Linux. См. также **Gnome** и **Xfce**.

Измученный своими же демонами и раскритикованный после выхода 4-й версии, KDE 4.7 предлагает ряд нововведений и исправлений, дабы умиротворить сообщество и создать стабильную среду обитания и тихую гавань для противников Gnome и Unity.

Plasma Workspaces – обобщающее название для технологий KDE типа Plasma Desktop, Plasma Netbook и Plasma Active и сердце всех продуктов линейки KDE 4 – претерпел множество изменений. И это не только исправления ошибок, но и новые функции для удобства пользователя.

Лидирует среди них обновленный интерфейс, ориентированный на использование на сенсорных и мобильных устройствах. *Konsole*, эмулятор X-терминала, больше не блокирует отсоединение USB-устройств для хранения данных, а аудиоконтроллер *KMix* улучшил поддержку PulseAudio.

Менеджер Bluetooth

KDE 4.7 может похвалиться экспериментальной поддержкой *Network Manager 0.9*, и теперь подключения Bluetooth управляются прямо с главной панели. Из всех ее элементов – часы, звук, менеджер беспроводного подключения и пр. – Bluetooth единственный, над которым не появляется описание при наведении на него мыши, но характерный значок Bluetooth говорит сам за себя.

Много хлопот в предыдущих релизах KDE доставляла интеграция приложений *GTK* с *Plasma Workspaces*, однако благодаря улучшениям тем *Oxygen GTK* они теперь прекрасно сосуществуют.



» KDE Software Compilation 4.7.0 — это Рабочие столы KDE Plasma, Приложения KDE и Платформа для разработчиков KDE.

Пожалуй, крупнейшая победа *KWin* – его производительность. Если старая версия оконного менеджера сильно тормозила, а после длительной работы еле ползла, новый *KWin* очень стабилен, даже при множестве активных окон. Помимо сильно почищенного кода, *KWin* обзавелся поддержкой OpenGL ES (см. врезку) и рядом других функций, улучшивших производительность.

Не в пример Unity, совместному детищу Gnome и Ubuntu, в KDE 4.7 вы можете насладиться всем богатством красок, не имея для этого специальной навороченной видеокарты.

Еще одна простая, но очень важная функция – добавление навигации к менеджеру запуска *Kickoff*. В прошлых релизах предусматривалась только синяя кнопка

«назад», при помощи которой можно было вернуться на один уровень во вложенном меню.

Со стрелками навигации в верхней части меню все стало значительно проще, особенно если необходимо отследить свой путь или пролистать назад несколько экранов.

Ну и со времени выхода KDE 4, ни один обзор не обойдется без упоминания Комнат.

Комнаты как они есть

Сурово раскритикованная и во многом непонятая опция Комнат занимает все более видное место на центральной панели и на рабочем столе.

Чтобы правильно использовать новые группы приложений в Комнатах, нужно понять, как они работают. Считайте каждую Комнату уникальной смесью рабочих пространств, виджетов и приложений. Каждая комната объединяет все активные элементы этих трех групп в одну рабочую область. Можно создать сколько угодно таких Комнат и держать в каждой свой набор приложений и виджетов для конкретной задачи.

То есть, переключаясь на другую Комнату, вы попадаете в своего рода иную рабочую среду: настройте браузер, текстовый редактор и электронные таблицы для своей «рабочей» Комнаты, а в Комнату «отдыха» поставьте музыкальные и видеоплееры и канал YouTube.

Поддержка OpenGL ES в KWin

Помимо вытравливания ошибок из *KWin*, оконного менеджера с композитингом, что добавило гладкости в пользовательский интерфейс, KDE 4.7 – первая версия, поддерживающая OpenGL ES 2.0.

OpenGL ES – подмножество 3D-библиотек для встроенных систем наподобие смартфонов и игровых приставок. Его поддерживают Android 2.2, телефоны Nokia и Samsung, BlackBerry Playbook и другие устройства.

Благодаря ему, KDE можно будет запустить во всей красе на мобильном устройстве, поддерживающем OpenGL ES 2.0, например, устройствах, базирующихся

на чипсете NVIDIA Tegra 2. В своем блоге Мартин Гресслин [Martin Gresslin], главный разработчик *KWin*, говорит, что поддержка OpenGL ES – важная веха на пути KDE к серверу Wayland.

Он также сообщает, что Kubuntu 11.10 будет поставляться с пакетом *kwin_gles*, позволяющим получить сборку с OpenGL ES без компиляции из исходных текстов.

Разработчики также трудятся над разделением кода композитинга и оконного менеджера, и к выходу KDE 4.8 надеются сделать библиотеку более модульной.

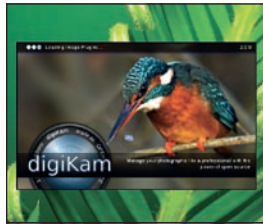


Свойства навскидку



Стабильный KWin

Улучшенный оконный менеджер KDE приобрел освещенный вид.



Обновления

Все главные приложения в коллекции ПО KDE пережили второе рождение.



➤ Менеджер Комнат позволяет группировать приложения, документы и виджеты.

Еще одна особенность данного релиза KDE – большое количество обновленных приложений в его коллекции ПО.

Akonadi – персональный информационный менеджер (ПИМ). Каждое ПИМ-приложение в семействе KDE 3 хранило информацию по собственному методу. С переходом к версии 4 KDE портировал все приложения на Akonadi. А общий механизм хранения и обработки данных означает лучший обмен информацией между приложениями. В релизе 4.7 в коллекцию ПО вернулся *Contact*, поскольку многие его компоненты уже портированы на Akonadi.

«KDE 4.7 предлагает среду для противников Gnome/Unity.»

Тестирование *KMail* показало многократные сбои при запуске, с ошибкой «Не удалось получить коллекцию из источника данных». Эта ошибка хорошо задокументирована и возникает из-за отсутствия у *Kmail* определения по умолчанию папки для входящей почты. В порядке временного решения можно определить в Akonadi *maildir* для *Kmail*. Для этого запустите Настройщик Akonadi и укажите для локальных папок путь `~/kde4/share/apps/kmail2`.



➤ Между релизами KDE 4.6 и KDE 4.7 разработчики исправили более 12 000 ошибок.

Dolphin, файловый менеджер KDE, также обновился и теперь выглядит аккуратнее. Вы заметите, что главное меню по умолчанию спрятано. Если без него вам пусто, нажмите `Ctrl+M`, и строка меню вернется.

Неротик – это средство взаимосвязи данных из разных программных приложений. В KDE его можно использовать для обмена метаданными между файлами и электронной почтой и т.п. Новый модуль расширения в версии 4.7 позволяет *Konqueror* и *Dolphin* задействовать Неротик, добавляя рейтинг к файлам и папкам.

Поклонники форматов CBR и CBZ обнаружат, что *Okular*, популярная программа просмотра файлов, теперь поддерживает просмотр директории в режиме комикса.

Gwenview, просмотрщик изображений по умолчанию, позволяет сопоставить два изображения друг с другом – очень нужная и популярная функция среди фотографов.

Ошибки отловлены

Работа над этой версией велась больше года, и речь идет не только о тех 219 исправленных ошибок, которые приведут нас к разговору о *Digikam 2.0*. Он теперь оснащен долгожданными функциями обнаружения лиц, улучшенным выделением и проставкой тэгов фотографий ит.д.

Лучшее автозавершение в Python, поддержка расширений LEX/YACC и *KDevelop* с поддержкой готовых стилей отступов – неплохой повод для радости разработчиков ПО. Также предлагаются обновленные *Kate*, *Marble*, *Kstars* и много других приложений.

При всем уважении к этим нововведениям, сопроводительные заметки, ключевая область KDE, требует более тщательной работы, внимания и планирования.

Для начала, KDE следует давать более подробные пояснения к разным своим известным технологиям. К примеру, в информации по релизу фигурирует Phonon, медиа-каркас, где появились такие новые

функции, как поддержка *Zeitgeist*, но эти новшества ни о чем не скажут человеку, который в принципе незнаком с этими технологиями.

Мы начали пробы KDE с Kubuntu 11.04, как с основной ОС, но потом перешли на OpenSUSE 11.4, поскольку столкнулись с неразрешимыми проблемами.

Установка KDE 4.7 на Kubuntu 11.04, которая изначально работает с KDE 4.6, прошла успешно. Но после перезагрузки мы столкнулись с двумя проблемами. И *Dolphin*, и KDE Daemon рухнули при попытке входа в систему. Отключение графических эффектов решило проблему с *Dolphin*, но KDE Daemon вылечить так и не удалось. Оказалось, многие пользователи сталкивались с той же проблемой, не только при обновлении до 4.7, но и в более ранних версиях.

На OpenSUSE, KDE 4.7 работал без особых помех, но это вовсе не значит, что он работал идеально. Самый раздражающий недостаток касается подсказок. В *LibreOffice* они выполнены черным текстом на черном фоне – и их не прочтешь. Наскоро проведенный поиск выявил те же проблемы у многих обновившихся до 4.7. Остается надеяться, что это будет быстро исправлено. **LXF**

LINUX FORMAT Вердикт

KDE 4.7

Разработчик: KDE
Сайт: www.kde.org
Цена: Бесплатно под GPL

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

» Лучший рабочий стол для тех, кто ценит время. Научитесь любить KDE 4.7, и он ответит вам взаимностью.

Рейтинг 8/10

Scientific Linux 6.1

Станет ли выход этого клона RHEL со странным названием последним гвоздем в крышку гроба CentOS? Следствие ведет **Маянк Шарма**.

Вкратце

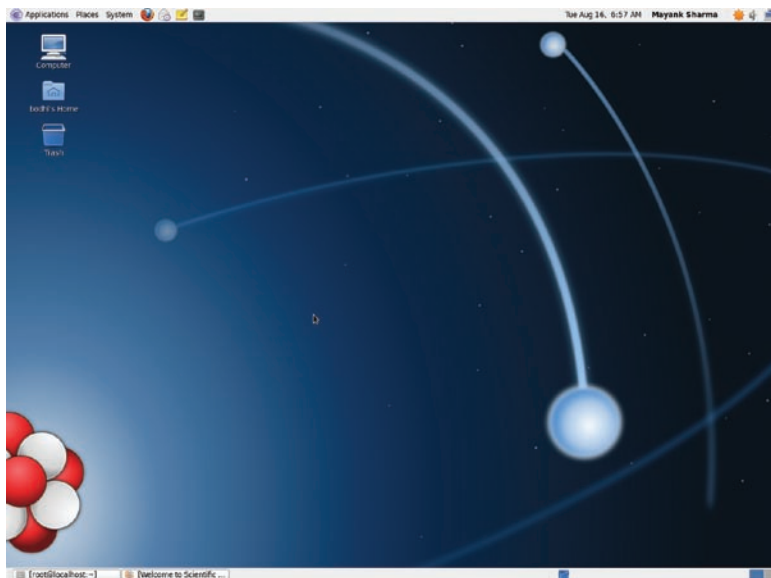
» Дистрибутив уровня предприятия, разработанный и используемый солидными организациями. См. также: CentOS или сам RHEL.

Этот дистрибутив был редким гостем на наших страницах: раньше его затмевал CentOS, знаменосец ниши для предприятий. Если в поисках серверного дистрибутива, по требованиям стабильности, надежности и репутации равного Red Hat Enterprise Linux, к вам обращались те, кто не имел средств на приобретение последнего, вы могли с чистой совестью указать на CentOS.

Но за последние пару лет все изменилось. CentOS до сих пор на 100% совместим с RHEL на уровне машинного кода, но отставание релизов распугивает пользователей.

И тут на сцену выходит Scientific Linux. Он также бьется за 100% совместимости кода с дистрибутивом-родителем, лишь кое-что добавляя от себя. Но последняя версия, 6.1, вышла всего два месяца спустя появления RHEL 6.1. Сравните с CentOS, который до сих пор в состоянии 6.0.

Пусть вас не смущает его «научное» название. Действительно, Scientific Linux был создан для людей в белых халатах, стремящихся проникнуть в тайны Вселенной в подземных ускорителях частиц. Согласно их сайту, этот дистрибутив изготовлен Fermilab и CERN совместно с некоторыми другими лабораториями и университетами, чтобы «уменьшить дублирование усилий... и иметь общую установочную базу для экспериментов».



» Поддержка Scientific Linux 6.1 будет вестись по крайней мере до конца 2017 г.

Но на самом деле дистрибутив не особо напичкан научными инструментами. Как и CentOS, это перекомпилированный из исходных кодов RHEL. Вытряхнув из пакетов RHEL все логотипы Red Hat, команда добавила ряд собственных пакетов. Среди них выделяются распределенная файловая система OpenAFS, применяемая во многих научных центрах, и облегченный оконный менеджер IceWM.

Для индивидуальной настройки есть утилиты `revisor` и другие средства создания респинов; правда, по умолчанию они не установлены. Важно отметить, что все это не уводит от класса предприятия. На практике, с этим дистрибутивом в CERN проводились эксперименты на Большом адронном коллайдере; вот вам наглядная иллюстрация его стабильности.

Верным путем

Есть и другие дополнения к родительскому дистрибутиву. Так, встроенный пакет `yum-autoupdate` позволяет автоматически устанавливать критически важные обновления.

Для дистрибутивов уровня предприятия жизненно необходимы обновления безопасности; разработчики и тут реагируют довольно быстро.

Номера версий установленного ПО называть нет смысла, поскольку они даже близко не совпадают с последними ре-

лизами. Ведь здесь на карту поставлена стабильность. Однако можно обратиться к сторонним репозиториям (см. врезку), чтобы добыть подключаемые модули и приложения.

Команда также выпускает спецпакеты, более известные как RPM для подстройки ('tweak RPM'). А конкретнее – это пакеты, подогнанные к работе с серверами без мониторов и способные отправлять консольный вывод и в последовательный порт, и на экран; а также пакет, меняющий `inittab`, чтобы тот запрашивал пароль `root` при загрузке в однопользовательском режиме. **LXF**

Удобство дополнений

Из-за требований совместимости с RHEL большая часть встроенных приложений сильно устарела. Но есть довольно много сторонних репозиториях, предоставляющих доступ к огромному количеству программ и приложений, чтобы вы могли настроить систему «под себя» и использовать ее как научно-исследовательскую станцию или полноценный рабочий стол.



» Многие сторонние репозитории имеют собственные каналы поддержки.

Для новичков есть ELRepo (<http://elrepo.org/tiki/tiki-index.php>) с библиотеками, позволяющими дистрибутиву работать с разной аппаратурой. EPEL (<http://fedoraproject.org/wiki/EPEL>) предоставляет такие пакеты, как `wine` и `yumex`, а также инструментарий для разработчиков оборудования. Есть также Repoforge (<http://repoforge.org>), RPM Fusion (<http://rpmfusion.org>) и ATrpms (<http://atrpms.net>).

Используя последние, имейте в виду, что часть пакетов может быть доступна во всех репозиториях, а это способно привести к проблемам с зависимостями и конфликтам, угрожающим стабильности основной установки.

LINUX FORMAT Вердикт

Scientific Linux 6.1

Разработчик: Fermilab и CERN

Сайт: www.scientificlinux.org

Цена: Бесплатно под GPL и другими лицензиями

Функциональность 9/10

Производительность 9/10

Удобство использования 8/10

Документация 8/10

» Идеален как дистрибутив класса предприятия, минус стоимость поддержки RHEL.

Рейтинг 9/10

VVVVVV

Эндрю Грегори помнит мирные времена, когда игры были не про стрельбу

Вкратце

» Назад к платформерам/головоломкам 1980-х. См. также *Braid*, *Frodo* + игры-оригиналы для C64.

VVVVVVV (то есть 6 'V') – платформер-головоломка, где вам предстоит путешествовать по игровому пространству отгадывая загадки, собирая предметы и избегая гибели на колючках и из-за привидений и других злодеев. Ну да, так и есть; но столь же вразумительно было бы описывать Мону Лизу как пригоршню краски на холсте.

Запуск VVVVVV вернет вас в золотой век игр C64/Sinclair Spectrum и заставит расплыться в широкой улыбке: эта угловатая графика и писклявые 8-битные звуки навевают столько воспоминаний, что вам почти наяву слышится шуршание лентопротяжки.

Сюжет игры в том, что вы – изображенный пикселями капитан космического корабля, который совершил аварийную посадку на планете с весьма причудливой гравитацией. А именно, силу притяжения можно менять, нажимая на пробел.

Вверх тормашками

Вот, собственно, и все: движение влево-вправо и смена гравитации. Никаких бонусов, никаких боссов в конце уровня и никаких голубых снарядов для сражений. На самом деле, раз уж нас так окунули в атмосферу ретро, стоит сказать, что в своей простоте местами VVVVVV больше напоминает *Tetris*.

Простота, однако, не означает примитивности. Из главного приема с изменением гравитации выжали все, что можно было придумать для головоломок по ходу игры.

Действие всякий раз разворачивается на одном экране, и его нельзя пере-



» На этом экране нужно инвертировать гравитацию, упасть на плитку на дне ямы, затем все перевернуть и оказаться наверху. Пустяки!

тать. А значит, вам никак не подсмотреть, что за опасности ждут вас впереди, пока вы на них не наткнетесь, и приходится то и дело гибнуть, рухнув на какие-нибудь нежданно-смертоносные-шипы. Этак недолго и затосковать; но игра размечена контрольными точками, с которых можно начать новую жизнь, не решая тех же головоломок заново.

Иной раз – например, если вы упали в тоннель с шипованными стенами – избежать смерти вам поможет не опыт, а моторная память.

Может, с годами наши рефлексы и правда замедляются, но пару раз мы бы решили загадку, если бы пошевеливались быстрее или потрудились подобрать очередную бирюльку.

Свободное падение

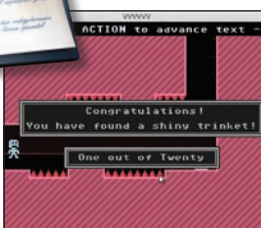
В роли капитана вышеупомянутого корабля, ваша миссия – отыскать пропавших членов экипажа, починить свое судно и покинуть планету.

Каждый член команды находится в ловушке в конце каждого уровня, и когда все они будут спасены, нужно вернуться на корабль. Однако сделать это не так-то просто: придется исследовать карту и определить уровни. Исследование заодно приносит бонусы, что придает вам ощущение контроля над своей пиксельной судьбой.

Возможно, лучшее, что есть в VVVVVV – это саундтрек, плод сумрачного гения Магнуса Полссона [Magnus Pålsson]. Он настолько хорош, что был даже выпущен отдельным синглом, а его ломаный электронный звук прекрасно дополняет графику.

Очевидным недостатком игры является ее краткость: мы все прошли за два часа. Но это были два захватывающих часа полной сосредоточенности и удовольствия. В те же £3,13 обойдется, скажем, билет в кино, но, в отличие от хлама, который крутят в вашем местном развлекательном центре, игра стоит затрат. **LXF**

Свойства навскидку



Блестящие бирюльки

Собирайте блестящие бирюльки, чтобы открыть дополнительные уровни к основной игре.



Карта

Нажмите Enter, чтобы увидеть карту. Она заполняется по мере вашего исследования пространства.

LINUX FORMAT Вердикт

VVVVVV

Разработчик: Terry Cavanagh и др.
Сайт: <http://thelettersixtim.es>
Цена: ок. \$5

Сюжет	8/10
Графика	10/10
Длительность	3/10
Цена	10/10

» Как игра и как произведение искусства, это один из лучших образчиков инди под Linux.

Рейтинг 8/10



Digikam SC 2.0

После года разработки, один из лучших инструментов управления фотоснимками достиг рубежа 2.0. Адам Оксфорд разузнал, что в нем новенького.



Вкратце

» Полное управление фотоснимками с поддержкой RAW и встроенным редактором. См. также: Shotwell и Picasa.

В чем главное достоинство рабочего стола KDE? С точки зрения приверженцев Gnome, это никак не Плазмойды или всякий там композитинг; это *Digikam*, менеджер фото, установленный в KDE по умолчанию.

В плане полноты набора инструментов для управления снимками с ним ничто не сравнится — только если пеершагнуть в проприетарный мир, где есть приложения вроде *Picasa* от Google. Но даже там *Digikam* не стущается — к примеру, за счет своего уникального умения подключать до 19 сервисов обмена фотографиями из главного окна.

Digikam быстр, гибко настраиваем, способен загрузить огромный объем снимков и пролистывать их по одному, не дожидаясь, пока в фоне догрузятся виртуальные библиотеки. А главное — он читает почти все существующие RAW-форматы и распознает их во встроенном редакторе быстрее, чем многие спецприложения.

Он также поддерживает 16-битное редактирование и полное управление цветом, что выводит его в первые ряды при-

ложений для управления снимками. Но это все уже было в версии 1.9; а что же появилось нового?

На первый взгляд, все как всегда. Это потому, что с точки зрения конечного пользователя нововведений не так уж много. Сам интерфейс, основная галерея и окна редактирования с виду остались прежними. Вместо этого разработчики сконцентрировались на устранении ошибок, а также на полезных — и по-настоящему уникальных — нюансах, таких как автоматическое распознавание лиц и редактирование геотэгов, при помощи щелчка на крутящемся глобусе. Практические нововведения, вроде быстрого доступа к закладкам, кажутся по сравнению с этим менее значимыми.

Но неуязвимым для критики *Digikam* не выглядит. По меркам стандартов фоторедакторов для любой платформы он более чем хорош, но не идеален. Смена версии могла бы послужить неплохим поводом улучшить интерфейс.

К примеру, значки, расположенные по бокам основного окна, плохо масштабируются к мониторам с высоким разрешением. Если разрешение больше 1280×1024, значки становятся мелкими, и на них неудобно нажимать. До ключевых инструментов редактирования трудно добраться: они до сих пор погребены в вы-

падающих меню в верхней строке окна. Подход *GIMP*, где все самое нужное расположено под рукой, подошел бы намного лучше.

Некоторые более новые функции тоже не мешало бы довести до ума. Конечно-му пользователю неясно, почему в одном окне геотэги можно редактировать перетаскиванием на глобус, а в другом — нет. И уж вовсе непонятно, почему кнопки ОК и Применить у разных инструментов расположены то вверху, то внизу.

Кроме того, практически невозможно найти готовую скомпилированную версию для 64-битной ОС.

Мелкие придирки

На самом деле все это скорее странности, нежели дефекты. Если мы и были чересчур строги в *Digikam*, то только для того, чтобы ему было к чему стремиться.

В работе со множеством изображений он по-прежнему безупречен, и использует для этого наиболее гибкую методику — объединение фотографий в коллекции и их отбор для редактирования. Удобство и простота в нем оригинально соединены с новаторством и профессионализмом.

Другими словами, у него очень широкие функциональные возможности, которые лишь изредка усложняют жизнь новичкам. И прежде всего, он стабилен. За исключением одной странности, когда программа то и дело возвращала нас к просмотру метаданных GPS, он надежен как скала, сколько фоток в него ни загрузи. И это, как ничто иное, делает его достойным почетного звания 2.0. **LXF**

«В работе со множеством изображений по-прежнему безупречен.»

Прогулка по Digikam

Панель инструментов

Почему-то просмотр то и дело переходит к виду миниатюр.

Инструменты управления

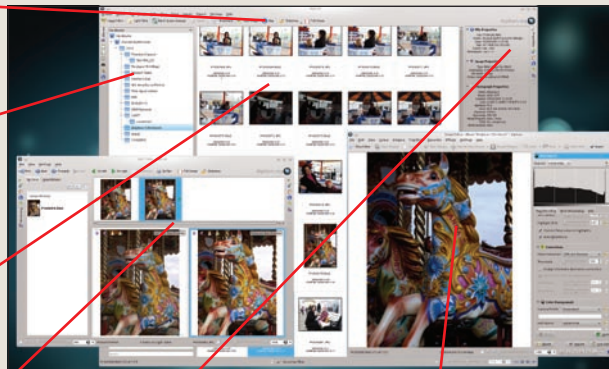
Можете просматривать по альбому, времени создания файла или тэгу, или запустить «умный» поиск, чтобы найти сходные изображения.

Главное окно

Переключается между миниатюрами и полным размером ваших снимков.

Монтажный стол

Поставьте рядом два снимка или миниатюры и выберите лучший.



Метаданные

Инструменты управления метаданными сидят слева, но чтобы изменить их, нужно будет открыть новое окно.

Редактор

Возможно, ему не хватает лоска, но встроенные модули *Kipi* предлагают множество эффектов, и поддерживается 16-битный цвет.

LINUX FORMAT Вердикт

Digikam Software Collection 2.0

Разработчик: Команда Digikam
Сайт: www.digikam.org
Цена: Бесплатно под GPL v2

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

Лучший инструмент OSS для управления снимками: стабильный, быстрый, полнофункциональный.

Рейтинг 9/10

Midori 0.4.0

Обновлен один из лидеров легковесных браузеров. Он остался за бортом нашего Сравнения, и **Игорь Штомпель** решил присмотреться к нему поближе.

Вкратце

» Легковесный и быстрый браузер на базе движка WebKit. См. также *Rekonq*, *Arora*, *Links*.

Как утверждает русская Википедия, Midori переводится с японского языка как «зеленый». В полном соответствии с этим толкованием, зеленый цвет нашел отражение в логотипе проекта (Меню > Параметры).

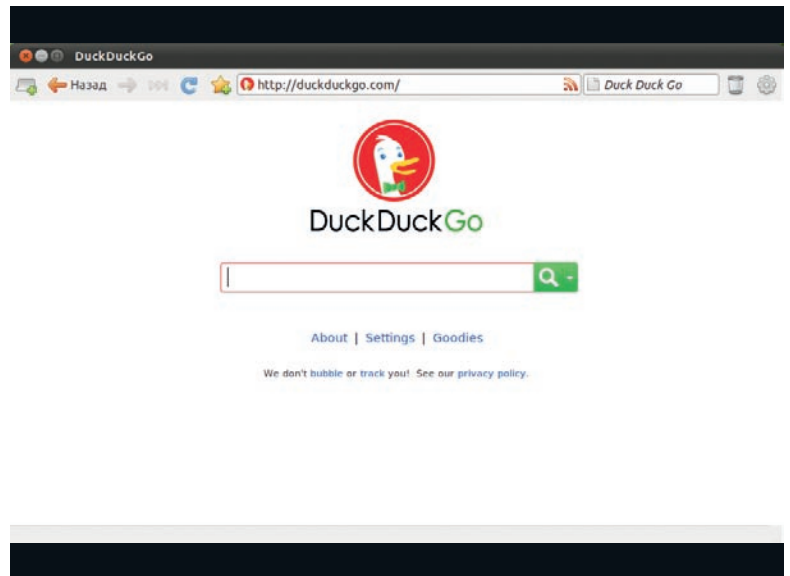
Первое знакомство

Midori основан на движке WebKit и полностью интегрирован с GTK+ 2, что предоставило возможность реализации многих популярных функций: вкладки, закладки, управление сессиями, панель быстрого доступа к поисковым системам, поддержка пользовательских скриптов и стилей, поддержка расширений. Кстати, откройте вкладку Дополнения (Меню > Параметры) – там вы сможете ознакомиться с доступными расширениями.

Что касается установки новой версии Midori 0.4.0, то она не вызвала у нас (Ubuntu 11.04) никаких проблем. Достаточно подключить репозиторий *ppa:midori* с Launchpad (<https://launchpad.net/~midori+archive/ppa>). После стандартной установки (для данного варианта) в главном меню в разделе Интернет появится пункт Midori.

Новшества

В Midori 0.4.0 реализовано достаточно большое количество изменений. Как и разработчики ведущих браузеров, создатели Midori уделили значительное внимание работе с JavaScript. В результате работа генератора JavaScript была улучшена. Но этим разработчики не ограничились. Так, была добавлена возможность



» Midori в действии!

передачи URI посредством вызова кода на JavaScript в командной строке (Меню > Информация о странице > Console). Среди других крупных новшеств – улучшение поддержки блокировки рекламы (всплывающих окон и т.д.) Adblock (например, поддержка правил игнорирования субдоменов), кэширование URL и поддержка Inline-парсинга.

Был усовершенствован и ряд диалоговых окон. Диалоговое окно, появляющееся после аварийного завершения программы, теперь позволяет отказаться от открытых в прошлый раз вкладок, вместо их загрузки открыв предыдущие страницы, а также открыть последние вкладки (соответственно, с загрузкой). Диалоговое окно Удаление личных данных (Меню > Удалить личные данные) получило новую опцию – Закрытые вкладки, которая позволяет очистить последние открытые вкладки. Из диалогового окна Новая закладка было удалено поле описания.

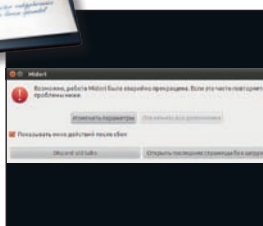
Отметим еще несколько интересных новшеств. Так, если вы пролистали текущую страницу до конца, можете перейти на следующую, просто нажав клавишу пробела (данная функция основана на обработке атрибута `rel="next"` или ссылок с названием Next). А просмотр исходного кода текущей страницы больше не приводит к повторной загрузке. Если понадобится прибегнуть к масштабированию,

не помешает иметь в виду, что разработчики сделали шаг масштабирования равным 10% (для возвращения к исходному масштабу можно воспользоваться сочетанием клавиш Ctrl+=).

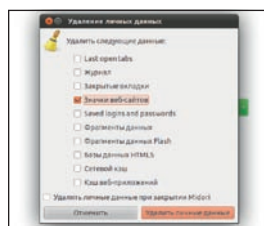
Новая опция “--log-file” (“-l”) позволит сохранить журнал в выбранный файл. Заодно улучшено и окно настроек.

Нам Midori понравился. Он быстр в работе, но самое главное – удобен и нетребователен к ресурсам. Документация у программы достаточно подробная (Меню > Часто задаваемые вопросы), но, к сожалению, на русском языке она не предусмотрена. **LXF**

Свойства навскидку



Аварийное завершение
После аварийного завершения Midori появляется новое диалоговое окно.



Очистка вкладок
В диалоговое окно очистки личных данных добавлена опция для очистки последних открытых вкладок.

LINUX FORMAT Вердикт

Midori

Разработчик: Проект Midori
Сайт: <http://www.twotoasts.de/>
Цена: Бесплатно на условиях GNU LGPLv2.1

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

» Легковесный и быстрый свободный браузер, удобный в работе, но без документации на русском языке.

Рейтинг 8/10

Сравнение

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

Web-браузеры

Мы все дольше засиживаемся онлайн, так что выбор браузера — дело важное. Мэтт Хэнсон сравнивает пять лучших предложений с открытым кодом.



Про наш тест...

Внимание следует обратить на две важнейших области: удобство работы пользователя и рабочая технология.

Тестирование технологий было относительно простым. Запустив браузеры в Ubuntu 11.04, мы проверили их возможности с помощью ряда эталонных тестов, включая стандартные SunSpider, Peasekeeper и Acid3. Это показало степень их совместимости с самыми современными web-технологиями и стандартами.

Однако 100-процентная совместимость с новыми стандартами и способность отображать JavaScript на пару миллисекунд быстрее соперника не делают браузер автоматически лучшим. Чтобы убедиться в удобстве пользователя, мы использовали тесты реального мира для оценки поведения каждого из тестируемых в повседневной жизни. Они включали посещение ряда различных сайтов, работающих на различных стандартах, включая новый HTML5.

Наша подборка

- » Chrome
- » Eriphany
- » Firefox
- » Opera
- » Rekonq

Как низко пали могучие. В самом начале того, что некоторые в слегка истеричной манере называли «войнами браузеров», двумя компаниями, претендующими на главенствующую роль в мире браузеров, были Netscape и Microsoft.

И хотя эта вражда, возможно, напрямую затрагивала только пользователей Windows, кончина Netscape и захват власти Microsoft в конце девяностых оставили пользователей Linux с полузаброшенными браузерами, которые иногда конфликтовали с web-страницами, разработанными исключительно под Internet Explorer.

Поражение Netscape привело к созданию Mozilla Firefox, чей идеал открытого кода дополнил собой смысл существования Linux. Его растущая популярность помогла уравнивать шансы на игровом поле и в итоге привела к отлично поддерживаемой альтернативе Internet Explorer, которую привечали web-разработчики.

Успех Firefox много значил для пользователей Linux, но недавняя серия неудачных и тупиковых решений, принятых Mozilla, привела к утрате популярности браузера Firefox в Linux. Одним из наиболее странных решений была отмена поддержки релиза после двух недель.

Привычка отмечать даже мелкие релизы номерами основных версий — кое-кто считает это нелепой попыткой изобразить такой же темп как у Explorer и Chrome — вызвала отчуждение некогда верных пользователей.

Итак, повеяло ветром перемен. С возвышением Google Chrome и ростом популярности и надежности таких альтернатив, как Rekonq и Eriphany, не пора ли забыть Firefox?

Web-браузеры определяют наш просмотр и взаимодействие с Интернетом, и именно поэтому так важно выбрать правильный.

Дополнения

Насколько расширяемы браузеры?

Коль скоро сам браузер должен быть стабильным и надежным, именно дополнения позволяют переключить его на ваш вкус, расширяя браузер и задействуя весь его потенциал.

Подавляющее большинство дополнительных создается третьими сторонами и энтузиастами; стоит ли удивляться, что самые популярные браузеры снабжены и самыми солидными и разнообразными подборками. У *Firefox* их множество, и они делают все — от блокировки рекламы и доступа в социальные сети до предоставления web-разработчикам инструментов серверной стороны. Многие из наиболее популярных портированы в другие браузеры.

Firefox нужна приличная система, позволяющая находить, скачивать и устанавливать нужные пользователям дополнения. Mozilla не разочаровала своим отличным сайтом (<https://addons.mozilla.org>), где дополнения перечислены по категориям и имеет облегченный доступ к популярным, полнофункциональным и наиболее перспективным дополнениям. Скачивание и установка полностью делается в самом *Firefox*, так же, как и обновления. Chrome Web Store (<https://chrome.google.com/webstore>) — еще один хорошо

продуманный ресурс. Быстрый рост популярности *Chrome* отражается в разрастании обширной библиотеки дополнений.

Opera традиционно придерживалась более закрытого подхода и не слишком охотно принимала дополнения третьих сторон. В версии 11 это изменено, и сейчас их количество на www.opera.com/addons постоянно увеличивается.

Немного порывшись, вы найдете нужное дополнение, хотя их выбор не слишком широк и некоторые представлены в старых, ошибочных версиях. *Opera* часто рассматривалась как новатор в области технологий браузеров — однако по части дополнений она пока что отстает.

Все усилия *Rekonq* тратятся на то, чтобы быть как можно легче, а значит, дополнения отходят на задний план. Правда, приложение для блокировки рекламы идет в установке по умолчанию, и предлагается поддержка дополнений на Flash и Java. Но пока *Rekonq* жертвует расширяемостью ради скорости.

И хотя *Epiphany* намного более открыт для дополнений, его небольшая пользовательская база обуславливает ограниченное количество официальных расширений, и они в основном ориентированы на произ-



» У *Firefox* по-прежнему самая крупная подборка дополнений.

водительность. Самые представительные из предложений *Epiphany* — *Certificate Viewer* и *GreaseMonkey*; здесь вы не найдете дополнений для Facebook.

Постоянный поток обновлений Mozilla часто оставляет дополнения позади. Предупреждение об этом весьма полезно, но может и огорчать. Ряд дополнений просто перестает работать с новыми версиями.

Firefox заработал репутацию крайне медленного браузера при установке чрезмерного количества дополнений. На скорость *Chrome* они не влияют так сильно, потому что он запускает дополнения в виде отдельных процессов.

Вердикт

- Firefox ★★★★★
- Chrome ★★★★★
- Opera ★★★★★
- Rekonq ★★★★★
- Epiphany ★★★★★

» *Firefox* пока на передовой, но *Chrome* быстро нагоняет его.

Безопасность

Основным приоритетом браузера должна быть ваша безопасность онлайн.

Похоже, Mozilla опять сама себя высекла своим отказом от поддержки предыдущих версий *Firefox*. *Firefox 4*, вышедший в этом году, уже не получает исправлений уязвимостей — и вынуждает пользователей переходить на *Firefox 5*. Новые функции, вроде опции Do Not Track, не позволяющей рекламодателям третьих сторон видеть ваши данные, показывают, что Mozilla еще способна на инновации.

Chrome предоставил инструменты создания песочницы — это лишний слой безопасности. Протокол Strict-Transport-Security (в *Firefox* для этого необходим модуль расширения) обеспечивает защищенное соединение с безопасными сайтами.

Opera применяет для безопасных сайтов шифрование SSL v3 и TLS, и, как и *Firefox*, уведомляет пользователя о безопасности сайта в адресной строке. Посещенные сайты также проверяются на предмет

их наличия в черном списке. Пользователей поощряют направлять информацию об обнаруженных ошибках или утечках в системе безопасно-

сти, и в среднем уязвимости исправляются гораздо быстрее, чем в *Firefox*.

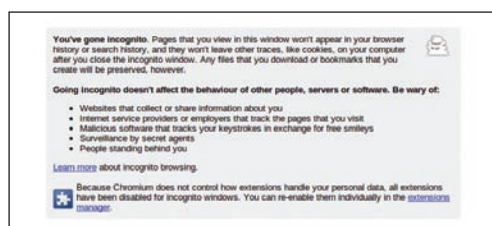
Прошлое *Rekonq* в смысле безопасности весьма неоднозначно: за последний год в нем выявились серьезные уязвимости. Но ситуация улучшается, и введено несколько неплохих функций, вроде виджета SSL, предоставляющего быстрый обзор идентичности, шифрования и безопасности по каждому посещенному сайту.

Epiphany практикует тот же подход, а еще в нем имеются всплывающий блокировщик и защита от cookie на ненадежных сайтах. Как и в *Firefox*, *Chrome* и *Opera*, вы можете сохранять пароли, хотя это менее надежно, чем в других браузерах.

Вердикт

- Firefox ★★★★★
- Chrome ★★★★★
- Opera ★★★★★
- Rekonq ★★★★★
- Epiphany ★★★★★

» И снова, похоже, Mozilla оказалась себе медвежьей услугой...



» Режим Инкогнито [Incognito mode] добавляет *Chrome* приватности.

«Прошлое *Rekonq* в смысле безопасности неоднозначно.»

Интерфейс

Может быть, внешний вид и не главное, но он определенно важен.

Интерфейсу браузера нужен баланс между предоставлением пользователю быстрого доступа к инструментам и соответствием своей основной цели – просмотру сайтов.

В прошлом браузеры радостно заполняли экран всякими кнопками и панелями инструментов – и каждое дополнение отрывало кусок бесценной площади экрана у отображения web-страницы.

После появления *Chrome* с его лаконичным интерфейсом другие браузеры начали избавляться от обильного визуального хлама, который и мешал просмотру страниц, и замедлял работу.

Однако аскетизм не должен вести к уродству, и привлекательный и элегантный браузер – вещь ценная, особенно когда речь идет о таком важном факторе, как удобство пользователя, о чем уже говорилось выше.

Firefox 5 ★★★★★

Firefox 5 поддерживает GUI предыдущей версии, и не без причины. До Firefox 4 интерфейс пользователя был разбухшим и перегруженным, что особенно раздражало при просмотре web-страниц с нетбуков с маленькими экранами. Урезанный интерфейс высвободил место на экране, сохранив легкий доступ к часто используемым функциям. Кнопки Вперед [Forward] и Назад [Back] ужалась, а кнопка Обновить [Refresh] тоже уменьшилась и переехала в адресную строку. Строка поиска предоставляет быстрый доступ к онлайн-поисковику, по умолчанию – к Google. Новые вкладки можно быстро открывать с помощью клавиши «+», а более продвинутые функции и настройки помещены в панель инструментов. В своей теме по умолчанию *Firefox*, возможно, выглядит не самым привлекательным из браузеров, но он практичен и эффективен.



Chrome ★★★★★

В дни, когда браузеры корчились под бременем огромного размера значков, бесполезных панелей инструментов и прочего хлама, появился *Chrome* – и стал пионером аскетичного интерфейса. В чем-то он подобен интерфейсу *Firefox* – или, лучше сказать, новообретенная страсть *Firefox* к простоте очень похожа на *Chrome*. Кнопки Вперед [Forward], Назад [Back] и Обновить [Reload] маленькие и ненавязчивые, но при этом легко доступны, располагаясь сбоку от адресной строки, она же – строка поиска, с единственным выбором – Google. Здесь нет кнопки Home – проявление безжалостной выбраковки кнопок Google. Панели инструментов тоже нет – доступ к продвинутым настройкам можно получить через значок с гаечным ключом. Очень опрятно, но не так просто в использовании, как у *Firefox*: порой кажется, что стремление к чистоте интерфейса затмевает цель простоты использования.

Воспроизведение мультимедиа

Мы говорим не только о противных файлах MIDI.

Интернет эволюционировал от сайтов с парочкой рудиментарных анимированных GIF и зацикленной MIDI-версией популярной песенки до действительно динамичного и интерактивного хостинга мультимедиа. Если ваш браузер не в силах правильно отображать мультимедиа, вы многое теряете.

Чтобы проверить, хорошо ли браузеры воспроизводят различные виды мультимедиа, мы использовали тест Futuremark Reasekeereg. Это серия тестов, оценивающих, среди прочего, качество рендеринга изображений, функций социальных сетей и сложной графики. Затем каждому браузеру

начисляются баллы – чем их больше, тем браузер надежнее и стабильнее в отображении сайтов.

На верхушке нашего чарта уселся *Chrome*, набравший в общей сложности 9537 баллов. И хотя особенно хорошо он показал себя в тестах с проверкой данных и текстов, он набрал 15841 за сложную графику, 5839 за социальные сети и 6960 за рендеринг. Второе место занимает *Opera* – в общей сложности 8603 очка. И хотя количество баллов, набранное в тестах с данными, было намного ниже, чем у *Chrome*, за сложную графику он набрал 18350 баллов, за социальные сети – 6384

и за рендеринг – 9131. *Chrome*, возможно, пригоднее в качестве повседневного браузера; *Opera* будет лучше для мультимедиа.

Epiphany занял третье место, прилично отстав от лидеров и получив 5636 баллов – 7248 за сложную графику, 3832 за социальные сети и 2996 за рендеринг. *Firefox* стал четвертым, набрав удручающе мало – всего 5524 баллов: хотя за сложную графику у него весьма достойная оценка – 14443 очка, но за социальные сети – 4009, и всего 3864 балла за рендеринг. И, наконец, *Rekonq* получил 5066 баллов: за сложную графику – 7711, за социальные сети – 3809, а за рендеринг – жалкие 2577.

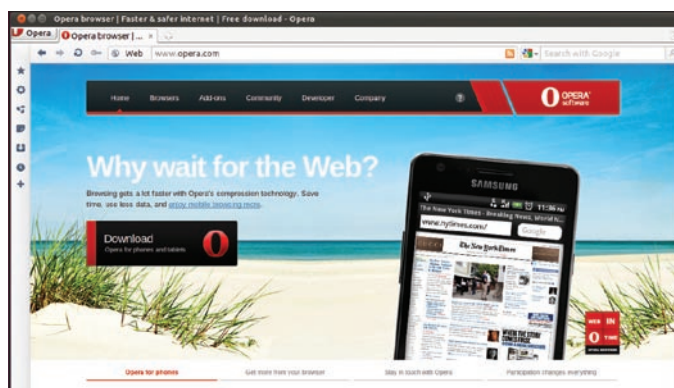
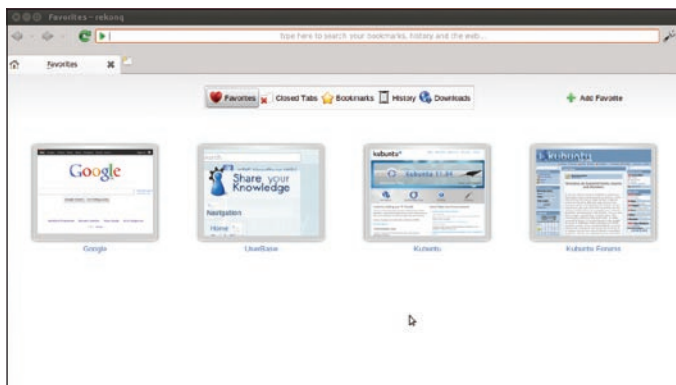
Вердикт

- Firefox ★★★★★
- Chrome ★★★★★
- Opera ★★★★★
- Rekonq ★★★★★
- Epiphany ★★★★★

» Rekonq отбросила назад слабая интеграция мультимедиа.

Opera ★★★★★

Opera сильно смахивает на подгламурную версию *Firefox*. Крупные кнопки и панели инструментов прошлых версий сменились компактными, аккуратными иконками. В отличие от всех других браузеров, здесь есть небольшая панель инструментов в нижней части окна с кнопками панелей – она добавляет с левой стороны еще одну панель инструментов: *Opera Link*, *Opera Unite* и *Opera Turbo*. Панель, бесспорно, удобна, но если вы не используете эти функции, то она просто съедает место на экране, где могло бы быть показано что-нибудь еще. Гораздо полезнее небольшой значок мусорной корзины в верхней правой части окна, дающий быстрый доступ к недавно закрытым вкладкам. Подход *Opera* к интерфейсу в чем-то противоположен *Chrome* – во главу угла ставится удобство, и вы получаете функциональный, хотя и немного сумбурный интерфейс.

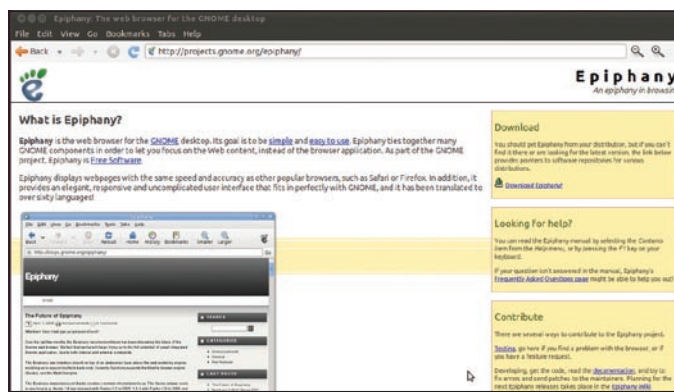


Rekonq ★★★★★

В определенном отношении *Rekonq* напоминает *Chrome*. Когда вы запустите этот браузер, перед вами предстанет страница с пиктограммами наиболее часто посещаемых сайтов. Из этой же страницы можно получить доступ к недавно закрытым вкладкам, закладкам, журналу и закамкам. Адресная строка здесь тоже совмещена со строкой поиска и допускает поиск в закладках и в журнале предыстории. Как и в *Chrome*, здесь отсутствует панель инструментов – предусмотрен только значок с гаечным ключом. И *Rekonq*, и *Chrome* оба базируются на движке *WebKit*, так что их внешнее сходство отнюдь не удивительно. Однако досадно, что *Rekonq* не сумел разработать столь же продуманный интерфейс, как у *Chrome*. Хотя интерфейс и урезан, все же непомерно много места отведено под кнопки и всякие-разные функции.

Epiphany ★★★★★

Epiphany выглядит возвратом к интерфейсу *Firefox* прошлых лет. Но Mozilla ушла вперед, и ее браузеры ориентируются на те же минималистские принципы дизайна, что и *Chrome*, а *Epiphany* сохранил крупные кнопки и панели инструментов. Иногда это неплохо – кнопки сразу же бросаются в глаза и на них легко нажать, а добавочные функции легко доступны и не спрятаны, как в *Chrome*, подобно постыдным секретам. Если вы используете *Epiphany* на настольном ПК, то площадь, отнимаемая кнопками и панелями инструментов, не будет такой уж неприятной проблемой. А вот на нетбуках ограничение экранного пространства для web-страниц немало раздражает. В общем и целом, интерфейс *Epiphany* незамысловат и напоминает работягу – не самый привлекательный, зато знакомый и простой в использовании.



Скорость

Быстрые страницы для мышелюбивых пальцев.

Мы немало платим за широкополосный Интернет, поэтому раздутый браузер с ошибками, который тормозит вашу работу в сети, означает пустую трату денег. К счастью, одним из лучших побочных эффектов перехода браузеров на минималистские интерфейсы стало уменьшение их ресурсоемкости.

Firefox 5 использует движок JägerMonkey на JavaScript, значительно увеличивающий скорость исполнения, и он показал среднее время 240,8 миллисекунд в тесте SunSpider JavaScript – серьезный отрыв от *Firefox 3.6* с его 1106 мс. Относительно

новая функция *Firefox* – обработка сложной графики частично передана GPU и видеокарте, благодаря чему работа в сети сделалась более плавной и более привлекательной внешне. При работе *Firefox* занимает 61,4 МБ памяти, хотя при установке дополнений этот показатель, естественно, растет.

Основной отличительной чертой *Chrome* является скорость, и его репутация как самого быстрого из основных браузеров вполне заслужена. Сайты открываются почти мгновенно, графика и анимация воспроизводятся быстро и без проблем. И результатом всего

этого становится быстрая и ровная работа в сети. *Chrome* выполнил тест SunSpider JavaScript за 272,4 мс, всего лишь чуть медленнее, чем *Firefox*. Однако это куда более легкая программа, требующая всего-навсего 30 МБ системной памяти – то есть намного меньше, чем *Firefox*. Именно принцип *Chrome* не сжирать ресурсов и делает его намного энергичнее конкурентов.

Opera пользуется преимуществами встроенной функции под названием Opera Turbo – внизу экрана помещается кнопка для быстрого включения и выключения этой функции.

Вердикт

- Firefox ★★★★★
- Chrome ★★★★★
- Opera ★★★★★
- Rekonq ★★★★★
- Epiphany ★★★★★

» И опять лидирует легковесный Chrome.

Функции синхронизации

Носите свои закладки всегда с собой, куда бы вы ни направились.

Firefox Sync встроена в версии, начиная с 4, быстра и проста в настройке. Информация защищена паролем, да еще и зашифрована; этой меры безопасности нет в *Opera*. Firefox Sync синхронизирует пароли, журнал, закладки и открытые вкладки – мы смогли продолжить сессию с другого компьютера.

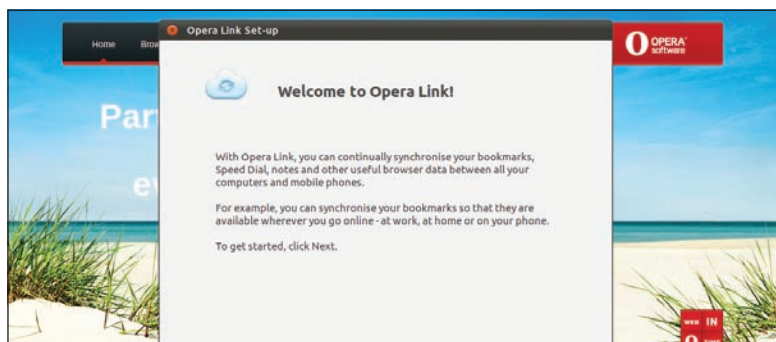
Функция синхронизации *Opera*, Opera Link, превосходно интегрирована и проста в использовании, хотя и не столь всеобъемлюща, как Firefox Sync. Вместо просмотра страницы Settings браузера вы можете получить быстрый доступ к функциям, щелкнув по значку в строке состояния.

«Opera Link отлично интегрирована и проста в использовании.»

Chrome использует учетную запись Google – вы ее получите, зарегистрировавшись на Gmail или Google+ – для хранения и синхронизации своих настроек. Как и в *Firefox*, информация защищена паролем и зашифрована. Также генерируется криптографический ключ, и *Chrome* предлагает ряд опций для его защиты.

Epiphany не имеет встроенных функций синхронизации, достойных обсуждения, но в нем есть ряд способов выполнить синхронизацию, например, дополнение Epsilon, сервис Ubuntu One, или через Dropbox.

Rekonq пока не имеет функций синхронизации, но в сообществе раздаются призывы реализовать их в будущем.



» Opera Link – отличный пример грамотной синхронизации данных.

Вердикт

- Firefox ★★★★★
- Chrome ★★★★★
- Opera ★★★★★
- Rekonq ★★★★★
- Epiphany ★★★★★

» Интеграция Opera Link завоевывает ей победу.

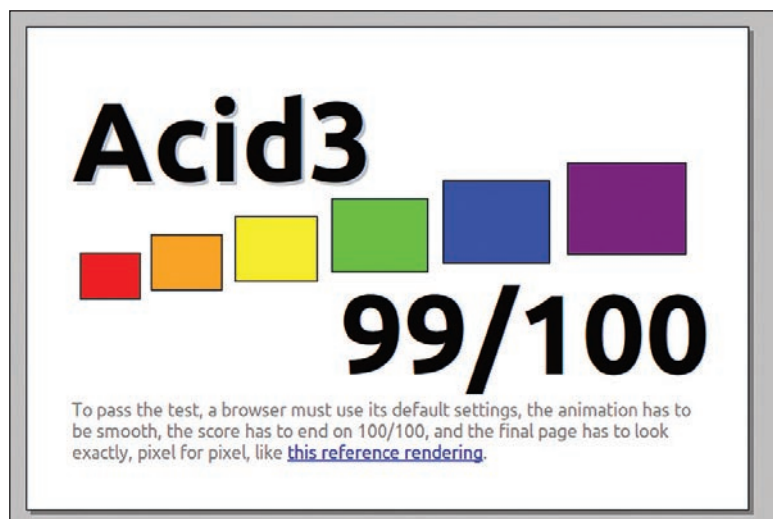
Совместимость со стандартами

Как браузеры справляются с различными web-стандартами?

Чтобы правильно отображать web-страницы, браузеры обязаны справляться с разнообразными технологиями и стандартами, применяемыми в Сети. Один из лучших тестов на совместимость со стандартами для браузеров – Acid3 (см. на сайте <http://acid3.acidtests.org>). Он тестирует соответствие браузера различным web-стандартам, уделяя особое внимание Document Object Model (DOM) и JavaScript. Чтобы пройти тест, браузер должен получить 100 баллов из 100.

Firefox не прошел тест Acid3, заработав 97 очков из 100 – неплохо, но все же это не полная совместимость с некоторыми web-технологиями. Предыдущий релиз тоже набрал 97 баллов; отсутствие сдвигов выглядит упущенной возможностью.

Из-за относительной новизны и быстрого взлета у *Chrome* есть проблемы с адекватным отображением страниц. Это в большей мере касается разработчиков и существенно только для старых сайтов. Популярность *Chrome* продолжает расти, и разработчики с этим считаются, так что проблемы должны прекратиться, или оправдания им не будет, поскольку тест



» Тест Acid3 проверяет совместимость браузеров с различными web-стандартами. Приемлем только идеальный результат – 100 баллов из 100.

на совместимость Acid3 *Chrome* прошел, набрав 100 баллов из 100.

Подобно *Chrome*, *Opera* прошла тест Acid3, и, с добавлением поддержки HTML5, былые проблемы с совместимостью в релизе 11.10 значительно снизились. *Epiphany* показал неплохой результат – 99 из 100 –

продемонстрировав большую совместимость, чем у *Firefox*. Поскольку *Rekonq* использует тот же движок рендеринга, что и *Chrome*, неудивительно, что он тоже набрал 100 баллов – хотя конечное изображение несколько отличалось от эталонного; у *Chrome* такой проблемы не было.

Вердикт

- Firefox ★★★★★
- Chrome ★★★★★
- Opera ★★★★★
- Rekonq ★★★★★
- Epiphany ★★★★★

» Завладеть тест Acid3 умудрился один Firefox.

Web-браузеры

Вердикт

Все протестированные здесь браузеры неплохи, и все имеют свои сильные стороны. Однако удивляет, что *Firefox* – когда-то считавшийся де-факто браузером сообщества открытого кода, да и вообще флагманским приложением – продемонстрировал слабость сразу в нескольких областях.

С учетом занимаемого им положения, такого мы не ожидали, но в наших тестах даже менее крупный браузер *Epirhany* превзошел *Firefox* по воспроизведению мультимедиа и безопасности.

Если *Firefox* не оправдал наших надежд, то относительный новичок *Chrome* превзошел их. Почти во всех тестах браузер Google занял лидирующую – или почти лидирующую – позицию. Он быстр и безопасен, совместим со всеми основными web-стандартами и справляется с воспроизведением мультимедиа. И даже на традиционном поле *Firefox* – в дополнениях – *Chrome* очень быстро догоняет.

Результаты *Opera* достойные, но не ослепляющие. Для большинства браузеров они сошли бы за счастье, но не для *Opera* с ее репутацией новатора на рынке браузеров: грустно видеть, что она почилла на лаврах, удовлетворившись возможностью плестись за кем-то, когда другие лидируют. Ее незначительное превосходство над *Chrome* в области цифровых развлечений свидетельствует, что этот браузер-ветеран всего удобнее для просмотра сайтов с избытком мультимедиа.

По сравнению с большой тройкой *Epirhany* кажется неудачником, но все же умудряется взять свое во многих областях. По части безопасности и совместимости со стандартами он показал себя очень неплохо, и хотя в плане скорости он отстает от других браузеров, он все же быстр и энергичен, и на него стоит обратить внимание, если вам нужен альтернативный браузер.



И, наконец, *Rekonq* не догнал другие браузеры в некоторых из проведенных тестов: но это не значит, что его надо полностью сбрасывать со счетов. Он еще очень молод, и его создатели выказали готовность прислушиваться к мнению сообщества и по возможности добавлять функции и поправки. С регулярно обещаемыми функциями этот браузер, возможно, еще не идеален, но за его развитием стоит наблюдать.

Ясно одно: *Firefox* лишился своей короны – да здравствует *Chrome*!

» **Надежность и достойная работа Chrome делают его явным победителем.**

«Firefox не оправдал наших надежд, а Chrome их превзошел.»

I Chrome ★★★★★
 Сайт www.google.co.uk/chrome Движок WebKit Версия 13.0.782.112
 » В большинстве наших тестов Chrome был лидером.

II Opera
 Сайт www.opera.com Движок Presto Версия 11.50
 » В борьбе за второе место Firefox и Opera шли голова к голове.

III Firefox
 Сайт www.mozilla.com Движок Gecko Версия 5.0.1
 » Несколько шаткая позиция для бывшего Номера Один.

IV Epirhany
 Сайт <http://projects.gnome.org/epiphany/> Движок WebKit Версия 3.0.4
 » Солидный web-браузер, который надо слегка отшлифовать.

V Rekonq
 Сайт <http://rekonq.kde.org> Движок WebKit Версия 0.7.0
 » Пусть и не самый лучший браузер, зато многообещающий.

Обратная связь

Вы все еще фанат *Firefox*? *Chrome* вам не хорош? Любите *Lynx*? Поделитесь своими мнениями по электронной почте: letters@linuxformat.ru

Рассмотрите также...

Исходный код *SeaMonkey* тот же, что и в *Firefox*, но разработка ведется сообществом, без вмешательства Mozilla Foundation. В нем по-прежнему имеются проблемы с совместимостью, но если вы все еще оплакиваете кончину Mozilla Application Suite, то *SeaMonkey* может вас устроить.

Midori – хороший выбор, если нужен легковесный и быстрый браузер, поддерживающий широкий диапазон web-стандартов.

В отличие от кое-каких более солидных браузеров нашего теста, *Midori* проходит тест Acid3 и снабжен всем необходимым для работы с большинством типов файлов и мультимедиа в Интернете, хотя стабильность и поддержка HTML 5 пока что не на высоте.

И, наконец, если вам нужен браузер в чистом виде, стоит обратиться к *Lynx*. Это текстовый браузер, который оставляет от сайтов только самую их суть. **LXF**

ДИСТРИБУТИВ СВОИМИ РУКАМИ



Джонатан Робертс показывает, как собрать дистрибутив, который делает именно то, что нужно для вас.



Дистрибутивов сейчас развелось немало — их сотни; но во всем этом многообразии иногда не отыскать такого, который точно отвечал бы вашим придирчивым запросам или потребностям экзотической ниши.

Ну да, если нужен суперстабильный сервер — для этого есть Debian или CentOS; кому подавай новое с иглолки ПО — берите Fedora или Arch; а для промежуточных вариантов всегда наготове Ubuntu.

А если вам требуется киоск, где выполняется всего одно web-приложение? Или — вы хотите распространять фото и видео в необщепризнанном формате?

Быть может, вам просто нужно, чтобы ваши любимые программы были установлены, настроены и готовы к работе сразу после

установки последнего обновления. А по умолчанию ни в одном из стандартных дистрибутивов этого нет.

Конечно, мы имеем дело с Linux, и оговорка «по умолчанию» весьма важна. Благодаря усилиям многих разработчиков стало доступным и даже несложным создать полностью свой

«А если требуется киоск, где выполняется всего одно web-приложение?»

дистрибутив, делающий все, что вы хотели.

На следующих страницах мы покажем три различные технологии сборки собственного дистрибутива. Мы не фанаты дублирования усилий, поэтому для каждой из них возьмем

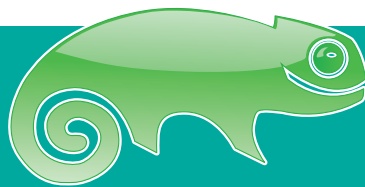
за основу другой дистрибутив. Это даст вам доступ к тысячам пакетов, без необходимости компилировать и организовывать пакеты самому.

Помимо того, что затрагиваются разные дистрибутивы, каждый способ направлен на свой уровень умения, чтобы сборка собственного дистрибутива была доступна и новому пользователю, и опытному профессионалу.

Начнем мы с самого простого, SUSE Studio, и дойдем в процессе до самого сложного варианта, переделки live CD Ubuntu.

В этой статье каждый найдет что-нибудь для себя; хотите — читайте ее от начала до конца, а хотите — просто останьтесь в более комфортной для вас части. Что бы вы ни делали, получайте удовольствие.

SUSE Studio



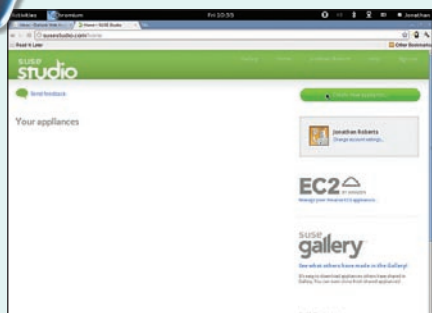
На этой и следующей страницах мы займемся SUSE Studio. Это уникальный и впечатляющий программный продукт, и с его помощью собственный дистри-

бутив на базе SUSE создается с легкостью необыкновенной, был бы только web-браузер. Надеемся, наша пошаговая инструкция даст вам представление о процессе и проиллю-

стрирует его простоту – перед тем, как вы перейдете в более продвинутый раздел и узнаете, как применять SUSE Studio с большей креативностью.



Шаг за шагом: Основы SUSE Studio



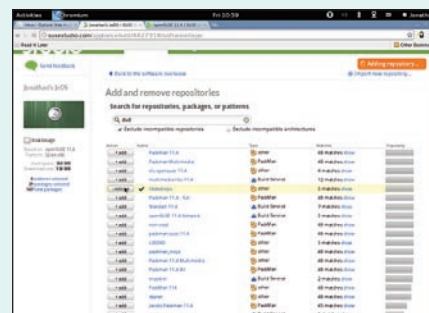
1 Создайте учетную запись

Войдите и выберите Create New Appliance. Ну что, вы, наверное, взволнованы? Ведь вы на пороге создания целиком и полностью своего собственного дистрибутива...



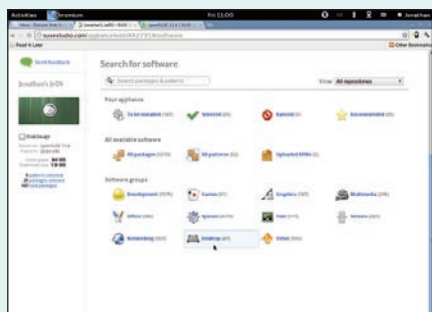
2 Выберите шаблон

Раз вы собираете собственный дистрибутив, шаблон берите минимальный. Не забудьте дать ему яркое имя, оригинальное и остроумное – допустим, JonSuse.



3 Добавьте репозитории

Перейдите на вкладку с программами и следуйте по ссылке, чтобы получить добавочные репозитории. Для настольной ОС рекомендуем добавить *Packman* и *libdvdrepo*.



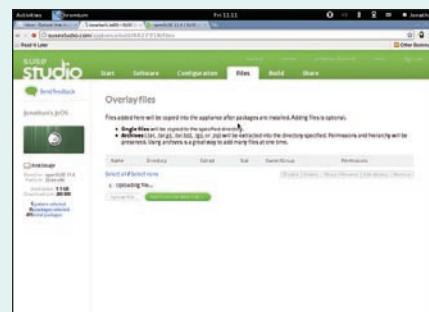
4 Добавьте программы

Добавьте желаемые пакеты программ, тщательно следя, чтобы не пропустить очевидные, но важные пакеты, такие как *NetworkManager*, *Gnome* или *KDE* и *wireless-tools*.



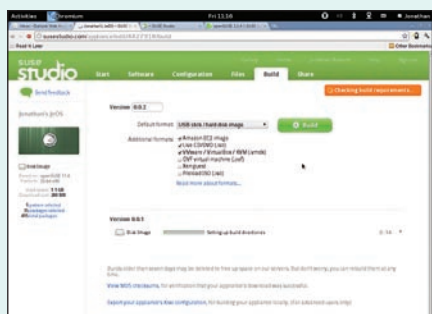
5 Опции локализации

Перейдите на вкладку настройки, укажите опции локализации под *General*, а чтобы не угодить после загрузки ОС в командную строку, задайте на вкладке *Startup* уровень запуска по умолчанию равным 5.



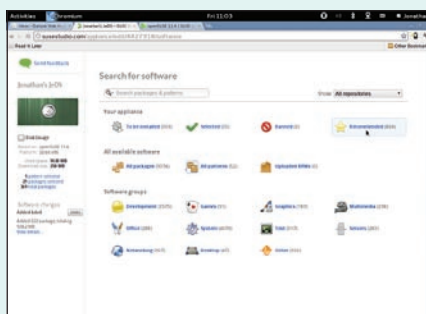
6 Загрузите файлы

Перейдите на вкладку *Files* и загрузите любые файлы, изображения или документы, которые вы хотите включить в ваш дистрибутив. Поместите их в домашнюю папку пользователя.



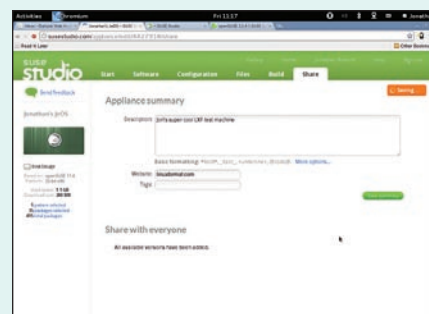
7 Выберите формат

В разделе *Build* выберите формат. Для Интернета подойдет *Amazon EC2*, для передачи друзьям – *USB-брелок*, а для экспериментов с виртуальными машинами возьмите *VMD*.



8 Сборка образа

Нажмите *build* и подождите. В *Chrome* можно включить уведомления рабочего стола, чтобы узнать, что все закончено, ведь сборка операционной системы требует времени!



9 Поделитесь со всеми

И, наконец, предоставьте свой дистрибутив всему миру на пробу. Зайдите на вкладку *Share*, снабдите дистрибутив описанием и любуйтесь его появлением в *SUSE Studio Gallery*.



SUSE Studio для асов

Представьте, что вы разворачиваете новую ОС в офисе вашей фирмы и хотите сделать домашней страницей браузера вашу интрасеть. Для сборки дистрибутива использовалась SUSE Studio, но там отсутствует опция для смены настроек отдельных приложений – там просто слишком много возможностей, чтобы реализовать их разумным образом в едином интерфейсе. Но выход есть, и узнав ответ, вы сочтете его простым.

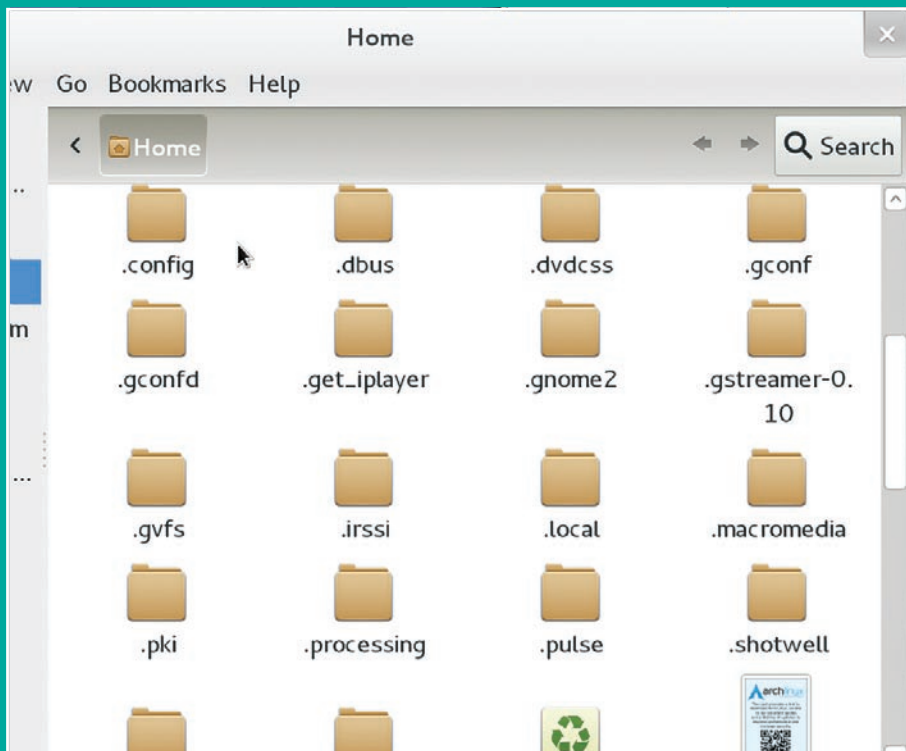
Все опции, которые пользователю-неадминистратору разрешено менять, содержатся в каталоге **home**. Обычно вы их не замечаете, так как они содержатся в специальных папках, по умолчанию скрытых – скажем, **.mozilla** и **.chromium**. (Если поместить **.** в начале имени любой папки, она станет скрытой. Команда **ls -a** или нажатие **Ctrl-H** в *Nautilus* отобразит такие папки.)

Так вот, все установки по умолчанию, используемые в приложениях, берутся именно из скрытых папок. Их лагерь – **/etc/skeleton**, и оттуда они копируются в каталог пользователя **home** при каждом создании нового пользователя.

Значит, чтобы сделать домашней страницей любого браузера пользователя интрасеть компании, нужно соответствующим образом видоизменить папки **.mozilla** или **.chromium** в **/etc/skeleton**. А в SUSE Studio это очень просто. Соберите дистрибутив, который вы хотите изменить, и вы увидите, что он появился внизу вкладки **Builds**, рядом со ссылкой *Testdrive*. Нажмите на нее, чтобы открыть виртуальный экзепляр вашего нового дистрибутива.

Когда он загрузится, вы сможете поменять домашнюю страницу, фон рабочего стола и вообще все что угодно. Это изменит папки с точкой в домашнем каталоге пользователя по умолчанию. После это следует взять их копии и добавить к вкладке **Files** в Studio.

Для копирования папки с одного компьютера на другой по безопасному сетевому соединению



Эти скрытые файлы прячутся повсюду, и они набиты опциями!

мы воспользуемся командой **scp**. В системе *Testdrive* от Studio по умолчанию сеть не работает; ее придется включить, нажав на ссылку над виртуальной машиной.

На той же странице можно увидеть пример команды **scp**, копирующей файлы на работающую виртуальную машину. Поскольку нам нужно, наоборот, взять файлы с этой машины на локальный компьютер, обратите внимание на значение, стоящее после опции **-p**, и что за номер у узла стоит после **root@**.

Обзаведясь этими данными, запустите терминал на локальном компьютере и выполните:

```
scp -r -o "UserKnownHostsFile /dev/null"
-p PORT_NUMBER root@nodeNUMBER.
susestudio.com:/home/USER/* /home/
LOCALUSER/temp/
```

В этом примере **PORT_NUMBER** должен равняться значению, стоящему после **-p**, **NUMBER** – номеру узла, **USER** – имени пользователя по умолчанию, установленного для виртуальной машины, а **LOCALUSER** – имени вашего пользователя. Это скопирует весь домашний каталог пользователя по умолчанию, чьи установки были изменены, в каталог **/temp** вашего каталога **home**.

После этого переключитесь на вкладку **Files** в Studio и загрузите туда содержимое папки, убедившись, что каталог назначения – **/etc/skeleton**. При следующей сборке вашего дистрибутива и запуске **testdrive** вы должны увидеть, что все изменения стоят по умолчанию, как если бы они были записаны на USB-брелок или загружены в виртуальную машину.



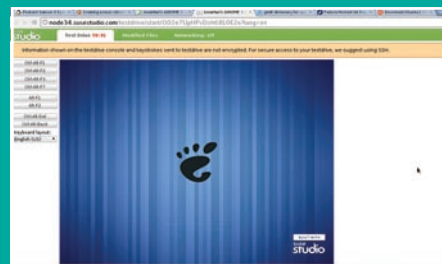
Скрипты в SUSE Studio

Раздел **Scripts** во вкладке **Configuration** – самая мощная функция настройки SUSE Studio. Здесь можно добавлять исполнение скриптов *Bash* в начало или конец сборки, а также в каждую загрузку вашего дистрибутива.

Использование *Bash* наряду со всеми остальными средствами, знакомыми пользователям Linux, открывает массу возможностей. Прекрасный пример на wiki-странице OpenSUSE – определение

графического оборудования при первой загрузке и установка подходящих репозитория и пакетов, которые включают ускорение графики.

Написание скриптов *Bash* выходит за рамки нашего урока, но нам хотелось наметить вам на имеющиеся возможности. Если вас интересует скриптописание, можем порекомендовать *Bash Guide for Beginners* и *Advanced Bash Scripting Guide*, оба они входят в проект Linux Documentation.



Режим testdrive в SUSE Studio – это виртуальная машина в вашем web-браузере. Она прекрасно подходит для проверки сборки и внесения изменений.



Личная Fedora

Из трех наборов инструментов, рассмотренных в этой статье, инструменты Fedora мы считаем интереснее всех. Они не только автоматизируют процесс сборки дистрибутива, но это еще и те самые утилиты, что команда Fedora применяет для сборки официальных версий. Лично нас это окрыляет: ведь, вооружившись терпением и сноровкой, можно создать дистрибутив не хуже Fedora.

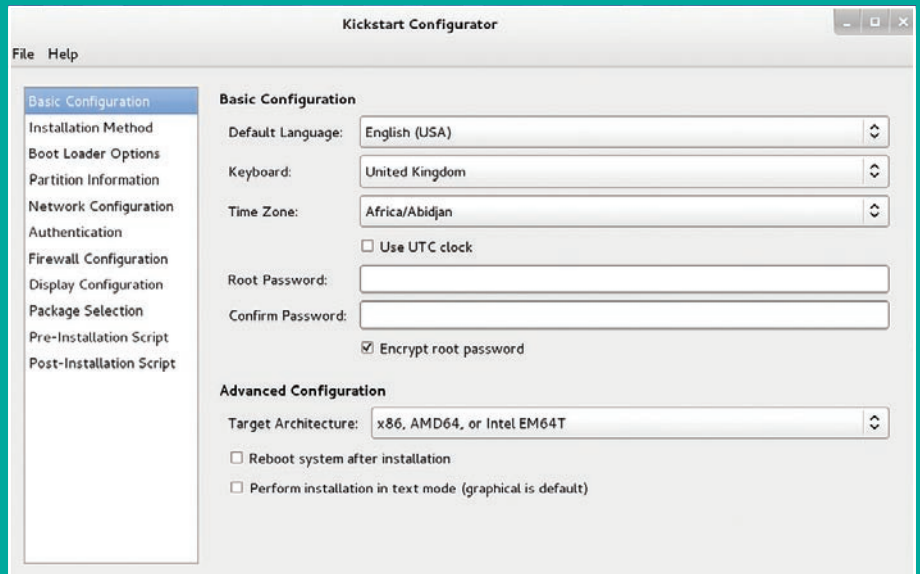
Утилиты для создания релизов в проекте – на выбор: *Pungi* только для установщика и *Live CD Creator* для создания live CD. Поскольку live CD очень удобны и популярны, мы здесь сделаем упор на них; однако в обоих способах используются файлы Kickstart, а значит, принципы создания одинаковы.

Kickstart

Прежде чем погрузиться в задачу создания собственного дистрибутива в Fedora, вам следует разобраться в работе *livecd-creator* и, что более важно, в технологии Red Hat, которая называется Kickstart.

Работа *livecd-creator* весьма похожа на ту, что проделывается при создании собственного live CD Ubuntu. Главное отличие в том, что здесь мы стартуем с пустой файловой системы, а не с заполненной заранее. Для заполнения этой файловой системы *livecd-creator* считывает инструкции из KS-файла – файла конфигурации, также известного как файл kickstart.

Такие файлы используются в *Anaconda*, программе-установщике Fedora, для автоматизации процесса и чтобы заранее ответить на вопросы, возникающие при установке. Разумно использовать их также и для создания live CD, поскольку, за исключением нескольких



► Пакет *Fedora system-config-kickstart* предоставляет графический интерфейс для создания файлов kickstart.

поскольку они выполняются сразу после парсинга файла, а разделы *packages*, *pre* и *post* – начинаться с объявлений *%packages*, *%pre* или *%post*, а заканчиваться на *%end*.

Мы сказали «как правило» просто потому, что в случае с live CD разделы *pre* и *post* интенсивно используются для создания новых пользователей, новых пакетов загрузки и так далее. Более того, файл kickstart может включать другие файлы kickstart, и порой трудно распознать, где какие установки работают.

Хотя оба эти фактора несколько усложняют создание собственного kickstart с нуля, но, на наше счастье, Fedora Project предоставил кучу файлов-примеров, пригодных в качестве основы для ваших новых проектов. Разделы *Includes* все еще остаются, но нам, по крайней мере, не придется возиться с *initramps!*

Мы покажем вам, как достать средства и образцы для нужных вам kickstarts, поможем осмыслить их и настроить для ваших собственных целей. Поскольку нам нужны официальные утилиты Fedora и их удобные примеры kickstart, первый шаг к построению дистрибутива на базе Fedora состоит в скачивании и установке этих пакетов. Выполните от имени root

```
yum install livecd-tools spin-kickstarts fedora-kickstarts
```

После этого вы увидите новую папку */usr/share/spin-kickstarts/* с 18-ю разными файлами kickstart. Для наших целей наиболее важным является *fedora-live-base.ks*: именно в нем определяется большинство сложных раз-

делов *%post*, которые необходимы для создания live CD.

Но главное, в нем создается пользователь для live CD, задается минимальный набор пакетов и настраиваются некоторые службы. Взгляните и убедитесь, что вы знакомы с этим – почитайте, в частности, комментарии; но не стремитесь производить в нем изменения, поскольку свои настройки мы будем выполнять в другом месте.

А именно, мы создадим еще один, новый файл kickstart, где все настройки и проделаем; но включим *fedora-live-base.ks*, чтобы гарантировать себе правильную работу. Потому-то он и важен для нас: он сильно упрощает жизнь.

Настройка

Чтобы начать работать с новым файлом kickstart, убедитесь, что мы сменили папку на */usr/share/spin-kickstarts/* и откройте от имени root новый файл для редактирования – назовите его подходящим образом, например *jonдора.ks*.

Чтобы поделиться своим kickstart, поместите в его начале комментарии, описывающие вашу новую версию Fedora, и предоставьте контакт для связи с вами по поводу исправлений или вопросов:

```
# jondora.ks
# Description: Jon's special blend.
# Maintainer: jonathan.roberts@futurenet.com
```



«С терпением и сноровкой можно создать дистрибутив не хуже Fedora.»

излишних дополнений, все, что происходит – это установка стандартной системы Fedora на сжатую файловую систему.

Файл kickstart – это, как правило, простой текстовый файл, содержащий

- » Раздел **command**, где определяются установки для всей системы, включая доступные репозитории.
- » Раздел **packages**, где задается, какие группы пакетов и отдельные пакеты устанавливать.
- » Разделы **pre** и **post**, где запускаются скрипты перед созданием корневой файловой системы и после этого.

Единственное требование к файлу kickstart – разделы *pre* должны идти самыми последними,

После этого включите файл **fedora-live-base.ks**, для гарантии, что все загрузится и будет работать:

```
%include fedora-live-base.ks
```

Вы заметили, что в файле **fedora-live-base.ks** в разделе **command** есть строки, где язык задается как **en_US**, а клавиатура – **US**? Эту установку по умолчанию можно отменить – просто поместите в свой **kickstart** другое объявление языка и клавиатуры:

```
lang en_GB.UTF-8
keyboard uk
```

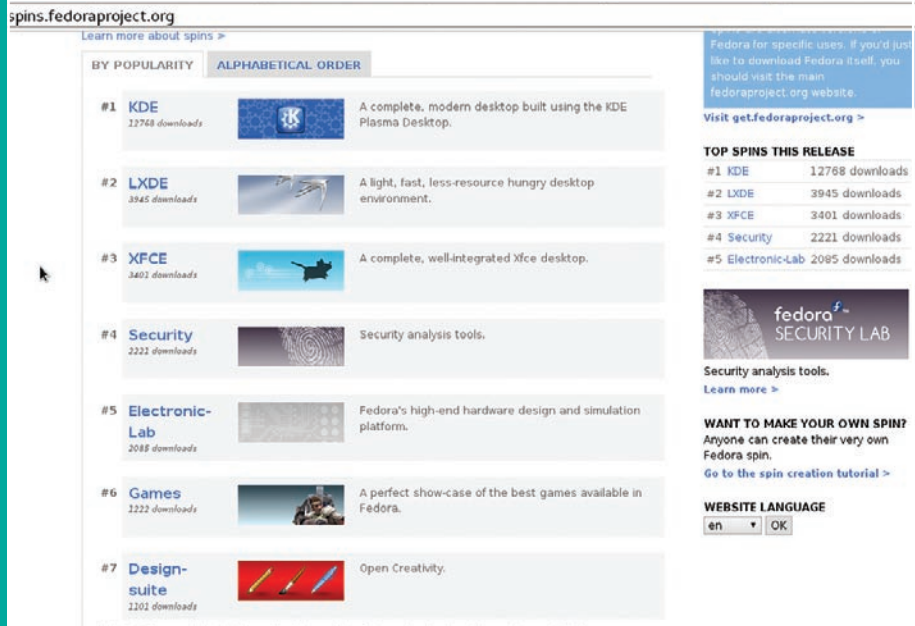
Оно сработает, так как файлу, который содержит включение, отдается больший приоритет.

Вам, скорее всего, захочется добавить дополнительные пакеты, включая те из RPM Fusion, что предоставляют поддержку воспроизведения мультимедиа. Просто добавьте новую строку с репозиторием в раздел **command**, содержащую имя и URL репозитория:

```
repo --name=RPMFusion --baseurl=http://
download1.rpmfusion.org/free/fedora/
releases/15/Everything/i386/
```

После этого, создав новый раздел **packages**, добавьте пакеты из любого из репозитория Fedora, а также RPM Fusion:

```
%packages
@gnome-desktop
gstreamer-ffmpeg
%end
```



► Fedora поощряет создание своих «отпрысков» и даже рекламирует их на официальном сайте.

Завершающая премудрость, которой мы поделимся – размещение файлов и настроек пользователя в его домашний каталог по умолчанию. Как и в Ubuntu, простейший способ – разместить их в каталоге **/etc/skel**, так как все присутствующие там файлы копируются

при создании нового пользователя. Более того, владелец файла, контекст SELinux и права доступа сбрасываются, и новый пользователь становится их владельцем.

Как и в SUSE Studio, простейший способ получить должным образом измененный каталог **home** – создать нового пользователя в существующей системе Fedora, войти под ним и произвести нужные модификации.

Чтобы эти файлы попали на пользовательский live CD, требуется добавить в файл **kickstart** раздел **%post**. Как правило, разделы **%post** выполняются в **chroot**, то есть в остальной части файловой системы хоста, и пользовательский каталог **home** там недоступен. К счастью, есть опция **--nochroot**, которая позволяет выполнять команды за пределами **chroot**.

```
%post --nochroot
cp -r /home/custom/* $INSTALL_ROOT/etc/
skel
%end
```

Мы задействовали переменную **\$INSTALL_ROOT**, так как **livecd-creator** в каждой сборке использует различные временные корневые файловые системы, и понять, какой каталог используется, каждый раз довольно затруднительно. Переменная **\$INSTALL_ROOT** всегда указывает на текущее расположение временной файловой системы, облегчая нам жизнь.

Итак, наша настройка закончена; осталось выполнить всего одну команду, чтобы собрать диск и пустить обработку файлов **kickstart**:

```
livecd-creator -c jondora.ks
```

Вот и все. Вы создали собственный live CD на базе Fedora, используя те же утилиты, что и команда, выпускающая релизы проекта. Ну не круто?

Товарные знаки

Хотя Fedora Project позволяет собрать вашу собственную версию его дистрибутива, он прекрасно осознает, как это может снизить способность защищать свои торговые марки. Поэтому не поленитесь ознакомиться с руководством по товарным знакам Fedora и убедиться, что если вы поставляете хотя бы один пакет не из официальных репозиториях Fedora, то все логотипы удалены. На первый взгляд – задача сложная, но проект,

понимая это, включил их в два пакета и предоставил замены, на которые не распространяется защита товарных знаков. То есть, если вы используете добавочные пакеты, все, что вам нужно – это добавить в раздел **%packages** такие строки:

```
-fedora-logos
-fedora-release
generic-logos
generic-release
```



► Пакет **generic-logos** от Fedora позволяет запросто удалить логотипы, но по умолчанию предоставляется только тема «Beefy Miracle», и вы, вероятно, захотите создать свою атрибутику.



Личный Ubuntu

Uubuntu – один из самых популярных дистрибутивов, и многие дистрибутивы являются производными от него. Вы вряд ли удивитесь, узнав, что и для вас существуют способы построить собственный дистрибутив на основе Ubuntu.

Создание собственной версии Ubuntu не так хорошо документировано, и здесь нет специально разработанного интерфейса, как в SUSE Studio или Fedora, так что это не для слабонервных. Тем не менее, вооружившись временем, терпением и следующими двумя страницами, можно достичь прекрасных результатов.

Выбор базы

На следующих двух страницах мы покажем вам, как взять Ubuntu live CD, влезть внутрь него, изменить пакеты и настройки, а затем записать все снова как образ загрузочного диска.

Так как мы основываемся на Ubuntu live CD, задача номер один – раздобыть ISO-образ текущей версии Ubuntu. Сказать это легко, но вам следует обдумать, какую именно версию брать: основную и более современную Standard или Long Term Support (LTS), более стабильную.

Скачайте ISO-образ с сайта Ubuntu. Следующий шаг – раскрыть его, чтобы получить доступ к его файлам. Можно записать образ на диск и скопировать содержимое на жесткий диск, но это лишний труд.

Лучше воспользоваться командой **mount** с опцией **loop**. Она велит Linux рассматривать

«chroot заставит компьютер думать, что верх иерархии системы — не /.»

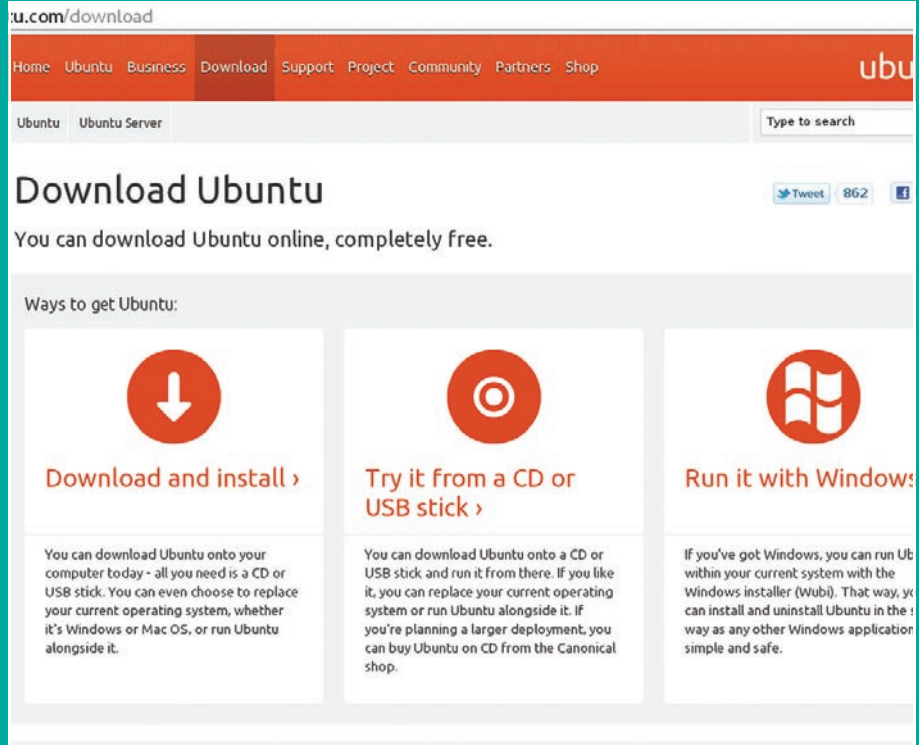
файл образа диска как физический CD, подключенный к системе.

Команда, выполняющую это действие, запускается от имени **root**:

```
sudo mount -o loop /PATH_TO_ISO/ISO_FILE.iso /mnt
```

Заглянув в каталог **/mnt**, вы теперь увидите содержимое ISO-образа; в качестве **/mnt** может выступать любая папка.

Это даст нам доступ к содержащимся на диске файлам, однако, как и для физического CD, эти файлы доступны только для чтения – изменить их мы пока не можем. Чтобы получить доступ на запись, их придется скопировать в другое, временное, расположение нашего жесткого диска. Создайте рабочую папку **work** (чтобы все было аккуратно) и подкаталог **cd_root**, и скопируйте содержимое ISO в подкаталог:



» Первое важное решение — делать ли собственный Ubuntu на базе стабильного релиза LTS или более передового Standard.

```
mkdir -p work/cd_root
rsync -r -a --exclude casper/filesystem.squashfs /mnt/* work/cd_root/
```

Мы исключили **casper/filesystem.squashfs**, потому что файловая система в Ubuntu live CD содержится в сжатой форме, ради экономии места. Для получения доступа к ней ее нужно разархивировать отдельным действием, и для удобства последующего

архивирования поместим все это в отдельную папку, не смешивая с остальным содержимым диска:

```
sudo unsquashfs -d work/root/ /mnt/casper/filesystem.squashfs
```

Теперь в каталоге **work/root** вы увидите все дерево каталогов Linux, и у нас есть доступ ко всем требуемым файлам.

Изменение диска

Нам предстоит работа по изменению содержимого диска, и мы сделаем ее с помощью команды под названием **chroot**. Она очень важна для создания собственного live CD Ubuntu и полезна для других целей, поэтому требует некоторых пояснений.

chroot заставит ваш компьютер думать, что верхом иерархии файловой системы является папка, отличная от **/** — при работе в **chroot** ваш

компьютер будет полагать, что папка **usr**, откуда выполняются все команды, расположена не в **/usr**, а, скажем, в **/home/jon/work/root/usr**, а домашняя папка пользователя будет тогда **/home/jon/work/root/home**.

По сути, это означает, что можно запускать **apt-get** и другие команды Ubuntu в системе, не являющейся Ubuntu, и они окажут действие на файловую систему live CD, а не хоста.

Теперь вам понятно, как это позволит изменить live CD, используя знакомые утилиты командной строки. Вам также необходимо знать, что остальная часть системы недоступна из **chroot**, и некоторые важные системные файлы и каталоги необходимо сделать доступными из **chroot**, примонтировав их внутри него. Вот они:

- » **proc** содержит информацию о работающих процессах, конфигурации оборудования и другую системную информацию.
- » **dev** предоставляет доступ к аппаратным устройствам системы.
- » **sys** содержит информацию ядра об устройствах и драйверах.
- » **/etc/resolv.conf** дает доступ к сети из под **chroot**.



Команды их монтирования имеют вид

```
cd work/root
sudo mount -t proc proc proc/
sudo mount -t sysfs sys sys/
sudo mount -o bind /dev dev/
cp /etc/resolv.conf work/root/etc/
```

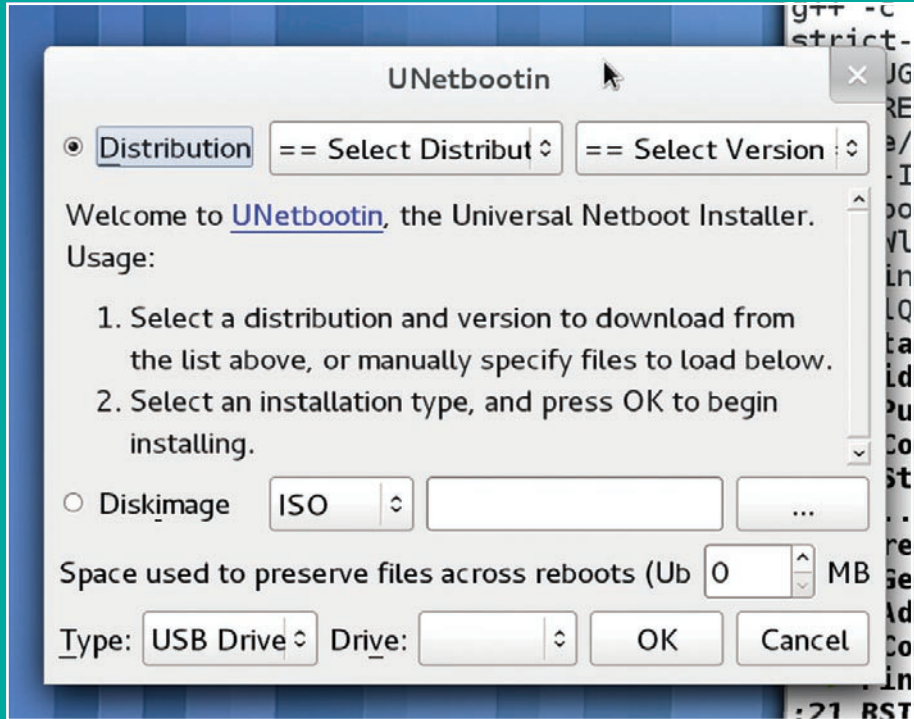
По завершении этих команд вы готовы вызывать **sudo chroot**, выполнять **chroot** в новую файловую систему и производить там какие бы то ни было изменения.

Изменения могут касаться включения по умолчанию репозитория `universe` и `multiverse`, что можно проделать, отредактировав `/etc/apt/sources.list`, и установки новых программных пакетов, таких как веб-браузер *Chromium* или мультимедиа-кодеки. Если вы хотите, чтобы в каталоге `home` по умолчанию появились добавочные файлы, просто скопируйте их, без помощи **chroot**, в каталог `work/root/etc/skel`: именно оттуда будут копироваться файлы при создании нового пользователя.

Пересборка ISO

Проделав все необходимые изменения, выйдите из `chroot` (командой **exit**) и отмонтируйте все каталоги (командой **umount**). Временные файлы, например, `/etc/resolv.conf`, тоже можно удалить, чтобы на live CD было больше места для значащей информации.

Прежде чем заново архивировать файловую систему `squashfs` и собирать ISO-образ с новой вместо старой, нужно еще поменять несколько файлов, чтобы предупредить о ваших реформах установщик Ubuntu *Ubiquity* и утилиту live CD, *Casper*. Вот эти файлы:



› *UNetbootin* запросто превратит ISO-образ Linux в загрузочный USB-брелок.

› `work/cd_root/casper/filesystem.manifest`, содержащий список всех пакетов, содержащихся в новой `squashfs`.

› `work/cd_root/casper/filesystem.manifest-desktop`, содержащий список всех пакетов, которые нужно установить — тот же самый список, кроме `ubiquity` и `casper`, поскольку их устанавливать не надо.

› `work/cd_root/casper/filesystem.size`, где записан размер файловой системы `squashfs`, который нужно знать установщику.

› `work/cd_root/md5sum.txt` гарантирует идентичность файловой системы той, что содержится на жестком диске в разархивированном виде.

Команд в этом разделе немало, и мы включили их на прилагаемый диск, чтобы сэкономить журнальную площадь и предотвратить ваши ошибки.

Проделав нужные изменения, пересоберите файловую систему `squashfs` и запишите ее обратно в папку `cd_root`, а затем загрузите ее в ISO-образ:

```
sudo mksquashfs work/root work/cd_root/casper/filesystem.squashfs
sudo mkisofs -J -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-loadsize 4 -boot-info-table -o myubuntu.iso work/cd_root
```

Удобные утилиты

Отныне вы можете проделывать с ISO-образом все, что захотите. Проверьте его в *Qemu*, используйте *UNetbootin*, чтобы записать его на флэшку, или делитесь им через сеть.

Стоит упомянуть, что навыки, полученные при создании собственного Ubuntu live CD, в особенности `chroot` и монтирование файловых систем, относятся к таким, которые линуксоид должен всегда иметь при себе. Если система рухнет, можете просто загрузиться с live CD, примонтировать вашу корневую файловую систему, выполнить **chroot** для нее, потом исправить ошибку — очень удобно. **LXF**

Ubuntu Customisation Kit

Создание собственной версии Ubuntu таким способом — процесс весьма трудоемкий. После того, как мы с вами прошли его от начала до конца и поняли принцип его работы, вам, возможно, будет приятно узнать, что есть утилита, автоматизирующая большую часть этой работы.

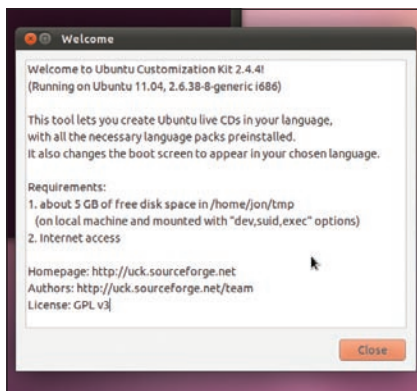
Она называется *Ubuntu Customisation Kit*, и доступна во всех основных репозиториях Ubuntu под именем **uck**. После установки и запуска вы увидите экран-приглашение, где объясняются возможности и требования программы.

К сожалению, этот экран хоть и полезен, но мог бы быть еще лучше (как и приложение в целом). Например, вам не говорят, что требуется ISO-образ той версии Ubuntu, которую вы хотите видоизменить — об этом вас спросят позже. Выбор языка и раскладка клавиатуры включает только сокращения, без описаний. Детали существенные, не так ли?

Когда вы проделаете основные шаги, вам представят выбор опций для настройки диска во время процесса сборки. Если ответить **Yes**, вы сможете использовать удобный графический менеджер пакетов или получить доступ к консоли, где

можно вызывать любые программы из командной строки.

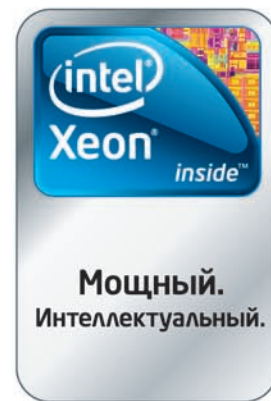
В итоге *UCK* выдаст вам ISO-образ, пригодный для любых целей. Это очень удобная программа, которая сильно упрощает жизнь.



› *Ubuntu Customisation Kit* предоставляет графическую утилиту для автоматизации многих этапов, рассмотренных в статье.

Как настроить сеть и не расстроить работу

Внедрение некоторых систем требует перестроения существующей сетевой инфраструктуры в соответствии со стандартами этих систем. Но это не относится к IBM BladeCenter® на базе процессоров Intel® Xeon®. Благодаря широкому спектру сетевых технологий, используемых в IBM BladeCenter, включая новейшие в отрасли решения для виртуализации, можно подобрать конфигурацию, которая лучше всего подходит для конкретной инфраструктуры. Кроме того, IBM BladeCenter позволяет сэкономить до 40% затрат на организацию сетей.



Реклама

Потратьте всего 10 минут – и убедитесь сами

Воспользуйтесь инструментом для оценки возможности консолидации систем и узнайте, как за 3 месяца окупить инвестиции в миграцию систем. Посетите ibm.com/systems/blade.ru

IBM, логотип IBM, ibm.com и BladeCenter являются товарными знаками или зарегистрированными товарными знаками International Business Machines Corporation в США и/или других странах. 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. Наименования других компаний, продуктов и услуг могут быть товарными знаками или знаками обслуживания третьих лиц. © 2011 IBM Corporation. Все права защищены.





Роб Пайк уже не новичок. Он работал над Unix в Bell Labs, приложил руку к созданию Plan 9 и даже засветился с Penn & Teller на *David Letterman Show*. Но его страсть — языки программирования, и последний его проект — Google Go: быстрый, параллельный, с открытым кодом. Майк и Грэм немного пообщались с Робом на недавней конференции OSCON и выяснили, как используется Go в Google и о чем вас могут спросить при найме в компанию.

LXF: Почему талисманом выбран суслик?

РП: Все дело в Go... [суслик — *англ.* gorper]. Пусть это глупо, но важно, чтобы у продукта был некий узнаваемый образ, ассоциирующийся с ним. Да он и неплох — людям нравится, суслики и правда популярны.

LXF: Что нового в Go за последние 12 месяцев?

РП: Ну, главная новость в том, что две платформы, не имевшие ранее поддержки, ее получили. Хотя вся работа была проделана сообществом Open Source, вне Google, теперь есть хорошая поддержка для Windows, а также App Engine. На Google App Engine это теперь третий язык, после Python и Java.

И теперь можно запускать приложения App Engine — пишете их на Go, а в облаке они будут собраны в статически скомпилированный бинарник, он и запустится. Скажем, если у вас идет какой-то ресурсоемкий процесс, вы можете использовать настоящую «числодробилку» для решения этой задачи, а мы сделали простенькую демо-версию, выполняющую кое-какие графические вещи, чтобы показать возможности. Кроме того, введено огромное количество исправлений и настроек.

LXF: Много ли вы работаете над Go? Или продвижением его внедрения в Google?

РП: Мы действительно используем Go (в Google), ведь у них изначально общее предназначение. Не берусь вдаваться в детали, но это относится ко внутреннему процессу.



ОБ ИЗМЕНЕНИЯХ

«Что мне действительно интересно — операционные системы уже не актуальны.»

LXF: Была ли его концепция связана с требованиями параллелизма, налагаемыми на язык?

РП: Да, хотя параллелизм — это только полдела. Вообще-то все началось с недовольства тем, что для написания серверного ПО приходилось использовать языки, которые работали плохо, по крайней мере, для нас. Стандартно, для всяких масштабируемых вещей применяют C++, но с ним куча сложностей. Параллелизм входит в наши планы, но это не самоцель. Лучшее — это скорость сборки. То же управление зависимостями реализовано таким образом, что пусть у вас множество деревьев — компиляция все равно будет очень быстрой, с чем C++ и Java (в меньшей степени) не справляются.

LXF: То есть главной целью было увеличение скорости компиляции?

РП: Да, для больших приложений. Стоит задаться этой целью, и вам потребуется новый язык, вот вы и начинаете думать — какой. Пока шла такая компиляция, мы и решили, что пора делать Go. Мы подумали: а не создать ли язык программирования с быстрой сборкой? Так что типичная сборка Go занимает долю секунды.

LXF: Да, при виде Go в действии больше всего поражает скорость.

РП: А сборка C++, даже для самой скромной программы — это много секунд. И Google держит эти огромные серверные фермы ради других ферм, и это маразм, никому они не нужны, а Go просто компилируется гораздо быстрее.

LXF: Полагаю, популярности ему прибавит App Engine?

РП: App Engine стал общедоступным на прошлой неделе (в середине июля), так что пока рано об этом говорить. Но, определенно, интересно будет посмотреть, что из него выйдет. Зарегистрировано уже несколько сотен приложений, использующих его.

LXF: После стольких лет в этой отрасли, ожидали ли вы, что языки будут так актуальны?

РП: Трудно сказать. Вот что мне действительно интересно, уходя в сторону от Go, так это что уже не актуальны операционные системы. Хотя многие от них в восторге, они уже не так важны.

LXF: Вы считаете это новым веянием?

РП: На самом деле, лет десять назад я давал интервью, где рассуждал об этом — что операционные системы, похоже, превращаются в пережиток прошлого. Все строят свое окружение так, чтобы оно либо было переносимым, либо работало на одинаковых API, а сама ОС уже роли не играет. Тем более, сейчас столько уровней виртуализации. Если вам понадобилось какое-нибудь приложение Windows, машина с Windows не нужна: поставили виртуальную машину — и дело в шляпе.

И то, что было лет 20–30 назад таким насущным вопросом — разработка операционных систем — уже не важно. Люди привыкли воспринимать их просто как один из уровней API, и поэтому потеряли интерес. С языками, по крайней мере, чувствуешь какое-то разнообразие. Раньше с операционными системами так тоже было, а теперь — нет. Возвращаясь к вашей

Роб Пайк

Грэм Моррисон и Майк Сондерс набираются программистской мудрости от одного из самых выдающихся инженеров.



му предыдущему вопросу – нет, я не думал об этом, но я не думал и о том, чем мне заняться вместо этого.

Когда я пришел в Google, компания была гораздо меньше, и одним из первых моих больших проектов был связан с новым языком программирования для высокопараллельного анализа данных. Мало кто верил, что легко создать жизнеспособный язык программирования. Но оказалось, что это неплохая идея, и она сработала хорошо. Мы спроектировали очень простой язык, позволяющий на один запрос задействовать несколько тысяч машин. Это тоже открытый код, хотя я не знаю, использует ли его кто-нибудь. Я-то его и не открывал, это уже другие постарались. Но проект был интересный. Go к этому языку никакого отношения не имеет. Go – не параллельный, а просто распараллеленный, но это был интересный эксперимент по продвижению языка вовне, а в самом Google сейчас функционирует с полдюжины внутренних языков для разных конфигураций и сетевого управления.

LXF: То есть не видеть нам одного языка на все случаи жизни?

РП: Нет. Похоже, будет языковой бум – после длительного периода, когда все пользовались только Java. Теперь же есть Erlang, и даже совсем древние языки переживают своего рода возрождение: Ruby, Closure, Scala и Pascal. Лет 10 назад о них не было ни слуху, ни духу. Теперь же они все потихоньку используются. Думаю, Go – из их числа, но у него поло-

жение особое, поскольку это по-настоящему компилируемый язык. Большинство других – интерпретируемые. Я считаю JVM интерпретатором, а вот Go компилируется в машинный код. Поэтому вы можете написать очень, очень быстрый код, сравнимый с C или C++. По правде говоря, я думаю, Go даже обходит C++. Ведь в том есть скрытые издержки, о которых людям свойственно забывать, а в C они очевидны.

LXF: Вы пользуетесь Go в обычной жизни?

РП: Безусловно. Пожалуй, я только им и пользуюсь. Я нахожу его очень эффективным, и многие со мной солидарны.

LXF: Считаете ли вы, что высокоуровневые языки лучше подходят для обучения будущих компьютерных специалистов? Или их следует также знакомить и с азами машинного кода?

РП: Думаю, здесь ключевое слово – «также». Вычислительной среде и так уже поднавредили, сводя ее к таким языкам, как Java и Python, где вы в упор не видите компьютера – только скрывающую его оболочку. Мы проводим собеседования, принимая людей на работу, и многие даже не знают, сколько занимает целое число.

LXF: Но нужно ли им это знать?

РП: Вам нужно это знать, если вы хотите писать настоящее ПО. Вот если вы хотите писать только связующий код, чтобы привязать страницу к вшей базе данных, тогда, пожалуй, нет.

LXF: Но не виноваты ли в этом языки программирования – что вам все еще нужно знать размер целого?

РП: Вы же не хотите терять 90 % процессорного времени из-за неверного распределения ресурсов, потому что вы не понимаете, как работает компьютер. Нехватка знаний дорого обошлась некоторым очень известным сайтам. Им приходится пересматривать все свои сборки, потому что слишком много вчерашних школьников не имели представления о производительности.

Высокоуровневые языки великолепны, пока в вашей программе не окажется миллион строк, при 100 программистах, над ней работающих, а вам невдомек, почему на выдачу одной страницы уходит 3 секунды. Дело не в том, чтобы всегда держать это в голове при написании кода, но если вы в этом не разбираетесь, то не поймете и где в коде это необходимо. **LXF**

Этапы развития

За последние 20 лет вышло в общей сложности 175 релизов ядра, не считая релизов безопасности с ныне устаревшей четвертой цифрой и нечетных «нестабильных» версий. Средняя частота – раз в шесть недель, в течение почти 20 лет. Релизы делятся на такие группы:

- » 0.x 13 релизов
- » 1.x 15 релизов
- » 2.0.x 41 релиз
- » 2.2.x 27 релизов
- » 2.4.x 38 релизов
- » 2.6.x 40 релизов
- » 3.x 1 релиз (пока)

Ключевыми релизами были:

- » Сентябрь 1992 г. первый релиз, 0.01
- » 14 марта 1995 г. 1.0.0
- » 9 июня 1996 г. 2.0.0
- » 26 января 1999 г. 2.2.0
- » 4 января 2001 г. 2.4.0
- » 18 декабря 2003 г. ... 2.6.0
- » 21 июля 2011 г. 3.0

При текущей схеме нумерации версию 4.0 можно ожидать летом 2021 года. Хотя здесь предполагается, что Линус не будет менять систему нумерации – гипотеза довольно шаткая.

Драйверы и поддержка устройств добавляются почти что в момент появления устройств на полках магазинов. От беспроводных карт до web-камер и даже Microsoft Kinect в Linux 3.0, диапазон поддерживаемых устройств огромен.

И это не только устройства, подключаемые к USB или PCI: возможность скомпилировать урезанную версию ядра лишь со строго необходимыми функциями и драйверами делает Linux идеальной ОС для встраиваемых систем. От мобильных телефонов и сетевых маршрутизаторов до автомобильных систем – везде есть собственные устройства, и ядро их поддерживает.

Диво-строки

Когда в ноябре 2010 года в новостях появился заголовок «200-строчная заплатка ядра Linux творит чудеса», в сети вспыхнули дискуссии и споры.

Задуманная для ускорения рабочего стола, эта заплатка разбивает задачи, запущенные на различных терминалах или в различных окружениях, на группы и делает так, чтобы ни одна группа не могла монополизировать процессор.

На практике это означает, что высокоинтенсивная фоновая задача, вроде компиляции программы (Линус, естественно, пробовал это на компиляции ядра) или перекодирования видео, не швырнет ваш браузер на колени – то есть дни перегрузки системы, с дерганием окон и срывами текста, ушли в прошлое.

Это интересно еще и потому (кроме самого факта, что столь важное изменение уложи-

лось всего в пару сотен строк кода), что ниче-го не требует от пользователя, если этот код включен в его ядро, то есть в любом дистрибутиве с версией ядра от 2.6.38. Этот код также распознает устройства любого типа, от сетевки на базе Atom до шестиядерного монстра.

С годами ядро существенно разрослось. Один из методов измерения объема программы – SLOC, число строк исходного кода [source lines of code]. Понятно, что этот показатель растет с каждым релизом, но общий рост буквально ошеломляет:

- » В версии 0.01 было 10239 строк кода
- » В 1.0.0 было 176250 строк кода
- » В 2.2.0 было 1 800 847 строк кода
- » В 2.4.0 было 3 377 902 строки кода
- » В 2.6.0 было 5 929 913 строк кода
- » В 3.0 – 14 647 033 строки кода

Да, все верно: ядро Linux выросло с 10 000 до 15 000 000 строк кода. С декабря 2003 года, когда вышло первое ядро 2.6, объем кода более чем удвоился. Сюда входит суммарное количество строк во всех файлах в архивах исходников, включая документацию. А поскольку большинство программистов считает написание документации гораздо более тяжелым трудом по сравнению с написанием кода, этот показатель выглядит достоверным.

Прежде чем кричать «Раздули!», вспомните, что Linux – не монолитное ядро: драйверы для большинства устройств поставляются в виде загружаемых модулей, которые в большинстве систем даже не нужно компилировать, не говоря уже об установке или загрузке. Рост размера исходников по большей части

происходит за счет все более полной поддержки устройств.

Одно из ложных представлений об открытом ПО, поощряемое его очернителями, состоит в том, что раз оно бесплатное – то в некотором роде любительское и худшего качества.

Анализ кода, написанного за 2009 год – это около 2,8 млн строк и 55 000 крупных изменений – показал, что три четверти изменений внесены профессиональными разработчиками.

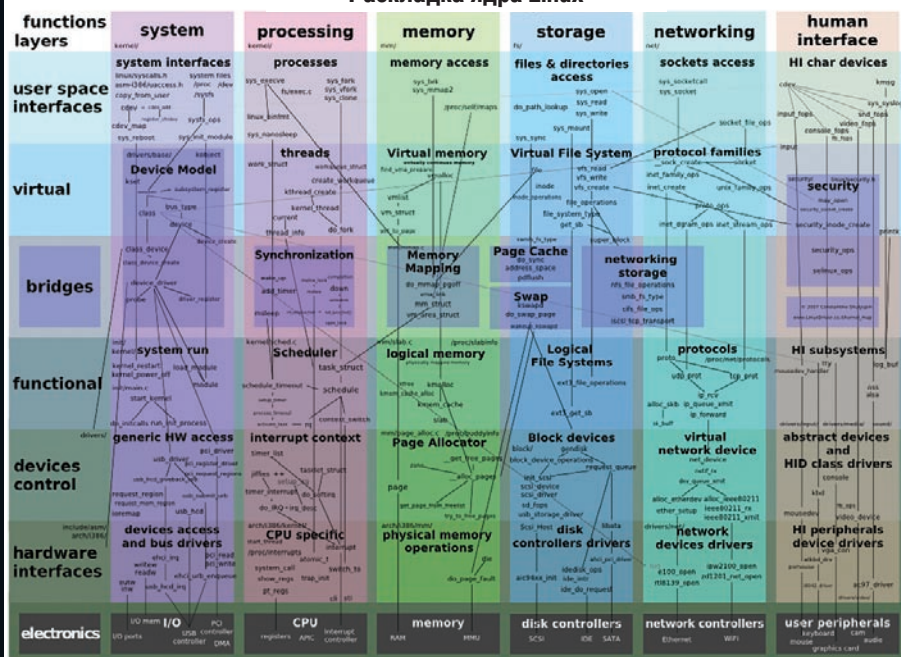
Неудивительно, что наибольший вклад внес Red Hat (12%), за ним следовали Intel (8%), IBM и Novell (по 6%). Несмотря на взаимную конкуренцию, эти компании также осознают важность взаимодействия. Каждая компания, конечно, разрабатывает области, прибыльные для себя, но от их работы выигрывают все.

Важность Microsoft

А вот что вас удивит – это что самый плодовитый индивидуальный поставщик кода в 3.0 работал на несколько иную компанию, Microsoft. Это отнюдь не означает, что компания «прозрела»: ей понадобилось вносить изменения в ядро, чтобы виртуальные машины Linux могли запускаться на ее платформе Windows Server. Таким образом, она стремилась не улучшить Linux, а добыть возможность продать побольше своих продуктов.

Какова бы ни была реальная причина, Microsoft теперь сложнее критиковать открытое ПО и ПО по лицензии GPL. И раз уж речь

Раскладка ядра Linux



» Не это ли Линус видит каждый раз, когда закрывает глаза?

»

зашла о Microsoft – ядро 3.0 также поддерживает кинетический контроллер Xbox, нашедший отличное применение во всех неигровых сферах.

Есть две вещи, которых в Linux множество: файловые системы и программы с непонятными названиями. Btrfs входит в обе категории, как ее ни называй: «Better F S», «Butter F S» или «B-Tree F S».

Это сложная файловая система, которая реализует ряд идей ReiserFS и ряд идей ZFS от Sun, а также обладает некоторыми функциями менеджеров логических томов, таких как LVM. Даже Теодор Цо [Theodore Ts'o], основной разработчик файловой системы ядра ext3/4, считает Btrfs шагом вперед.

Когда Btrfs впервые ввели в ядро (версия 2.6.29), она считалась во многом экспериментальной, но с тех пор созрела, и Fedora объявила, что они хотели бы сделать ее файловой системой по умолчанию в Fedora 16.

Будет ли так на самом деле или обождет до версии 17, это все равно свидетельствует о зрелости файловой системы, хотя она все еще находится в активной разработке, и обновления появляются в каждом релизе ядра.

Столь привлекательной Btrfs делает ее масштабируемость, девиз которой – «пусть Linux масштабируется на любую доступную систему хранения данных». Это означает не только большие диски, но и работу с несколькими дисками, и, что всего важнее, простоту и прозрачность этой работы для приложений. Добавочные возможности, такие как прозрачное сжатие, интерактивная дефрагментация и снимки, делают ее еще привлекательнее.

Обещание выполнено

Еще в 2006 году появился сайт, провозглашавший целью создание открытых 3D-драйверов для видеокарт Nvidia. Многие из нас заинтересовались этой идеей, но ожидали, что она уйдет не дальше, чем многие проекты на Source-

forge. Иногда приятно ошибаться в прогнозах! В ядре Linux теперь есть драйверы Nouveau для видеокарт Nvidia.

Если вы собираете собственное ядро – они находятся в области подготовки конфигурации ядра. В этой области, отключенной по умолчанию, находятся более экспериментальные или нестабильные (в смысле подверженности изменениям, не обязательно нестабильной работы) драйверы. Однако во многих дистрибутивах эти драйверы теперь входят в версии ядра и установщики по умолчанию.

Хотя они не дают той же производительности 3D-графики, как собственные драйверы Nvidia, их более чем достаточно для большинства неигровых применений, и они позволяют избежать проблем, способных появиться при смещении проприетарного, скомпилированного кода с открытым.

Вряд ли эти драйверы, полученные в результате обратной инженерии, когда-либо обгонят те, что написаны людьми, обладающими глубинными знаниями о работе устройств

«Споры позволяют обосновать выбор включаемого в ядро кода.»

и имеющими доступ к коду, которым Nvidia пользуется, не разглашая его. Однако они гораздо лучше априорных ожиданий большинства людей, и часто стоит немного пожертвовать скоростью, которая вам никогда не пригодится, ради надежности «родного» драйвера.

AppArmor (Application Armor – броня приложения) – модуль безопасности ядра. Он позволяет ограничить возможности отдельных программ посредством профилей. Назначение профиля программе или набору программ определяет допустимые для них действия. Это означает, что даже если для запуска программы необходимы привилегии суперпользователя

(root), она не сможет делать все, что захочет – это ограничение поверх стандартной системы доступа на основе прав пользователей.

Теперь недостаточно, чтобы запускаемые пользователем программы были «доверенными». Программе выдаются явные права доступа, чтобы она не выходила за рамки того, что ей нужно. В Linux уже есть SELinux, решающий похожие задачи, но AppArmor в целом считается проще для изучения и использования. Это важно, так как сложные системы безопасности труднее настроить правильно, и система безопасности, которая не защищает вас, когда вы уверены в обратном, наносит больше вреда, чем полное отсутствие защиты.

Рожденные в пламени

AppArmor создан уже давно. Его предыдущее воплощение, известное как SubDomain, появилось в Immunix Linux в 1998 году. В 2005 он был выпущен под названием AppArmor компанией Novell и включен в OpenSUSE и SLES. Но только в октябре 2010 года, начиная с релиза 2.6.36, он вошел в ядро как встроенный его компонент.

Если вы хотите почитать о тонкостях разработки ядра, есть только одно место, куда пойти: LKML (Linux Kernel Mailing List – рассылка ядра Linux). Здесь кипит работа и важные дискуссии, но это место не для слабонервных. Трафик довольно большой, около 200–300 писем в день, причем глубоко технических. Также стоит сказать, что многие разработчики известны благодаря способностям скорее к написанию кода, чем к общению.

Да и общение по электронной почте подразумевает более резкие манеры по сравнению с личным – «галантности в Интернете никто не поймет» – плюс фактор «эго» при обсуждении достоинств или недостатков собственного детища; получается среда, которая скорее продуктивна, чем дружелюбна.

Споров хватает, но они служат определенной цели, позволяя разработчикам обосновать выбор включаемого в ядро кода. Когда дебаты касаются тем, а не личностей, они продуктивны. Линус признал, что в некоторых из этих обсуждений, хотя и не слишком часто, даже он изменял свое мнение, поняв, что его исходная позиция была неверной.

За последние годы было несколько значительных споров/дебатов: например, стоит ли включать в ядро двоичные прошивки (проприетарный код, необходимый некоторым драйверам для взаимодействия с устройствами) – тема, вызывающая подскок давления у юристов свободы ПО и привлекательная для прагматиков.

Или – спор между Линусом и разработчиками ARM об их в чем-то изоляционистской позиции. Он разрешился удачно, и большой объ-

Линус Торвалдс

О разработке ядра невозможно говорить, не упомянув о Линусе Торвалдсе [Linus Torvalds]. Он не только начал этот проект и дал ему свое имя – он ведет его уже 20 лет и, кажется, не собирается останавливаться.

Его сообщение в Usenet, с которого все началось 20 лет назад, гласило: «Я делаю (бесплатную) операционную систему (это просто хобби, она не будет большой и профессиональной, как GNU) для клонов AT 386(486)».

Это «хобби» заняло у Линуса все время большей части его взрослой жизни. Он также довольно быстро избавился от недостатка амбиций. Зна-

чально не собираясь создавать «большую и профессиональную систему», релиз 1.0 он снабдил комментарием о том, что его целью является глобальное мировое господство – и даже читал лекции на тему «World Domination 101» (Мировое господство 101).

Это было шуткой, но он не часто вспоминает ее сегодня, потому что Linux становится значительной частью современных операционных систем уже вполне серьезно.



ем кода, относящегося к ARM, был перенесен в «основное древо», а не укрылся в своем углу. А так как сегодня устройств с Linux на ARM (предпочтительный процессор для встроенных систем и смартфонов), пожалуй, больше, чем на x86, это решение было благоразумным – даже если (или – потому что) родилось в пламени споров.

Виртуализация проникла повсюду, от компьютеров читателей *Linux Format*, пользующихся *VirtualBox* или *KVM* для тестирования дистрибутивов с LXF DVD, до крупных информационных центров, предоставляющих хостинг на виртуальных машинах.

В ядре есть поддержка систем виртуализации *KVM* (*Kernel-based Virtual Machine* – виртуальная машина на основе ядра) и *Xen*. Также всесторонне поддерживаются проприетарные системы, такие как системы от *VMware* и *Microsoft*, поэтому виртуальные машины Linux можно запускать почти везде.

Виртуальная машина эмулирует устройства компьютера, поэтому все, что может уменьшить объем программной эмуляции и разрыв между виртуализированными устройствами и «железом», на которых они работают, имеет значение. Расширения гипервизора *KVM* за некоторое время сделали это для процессора, и теперь то же самое производится с сетью.

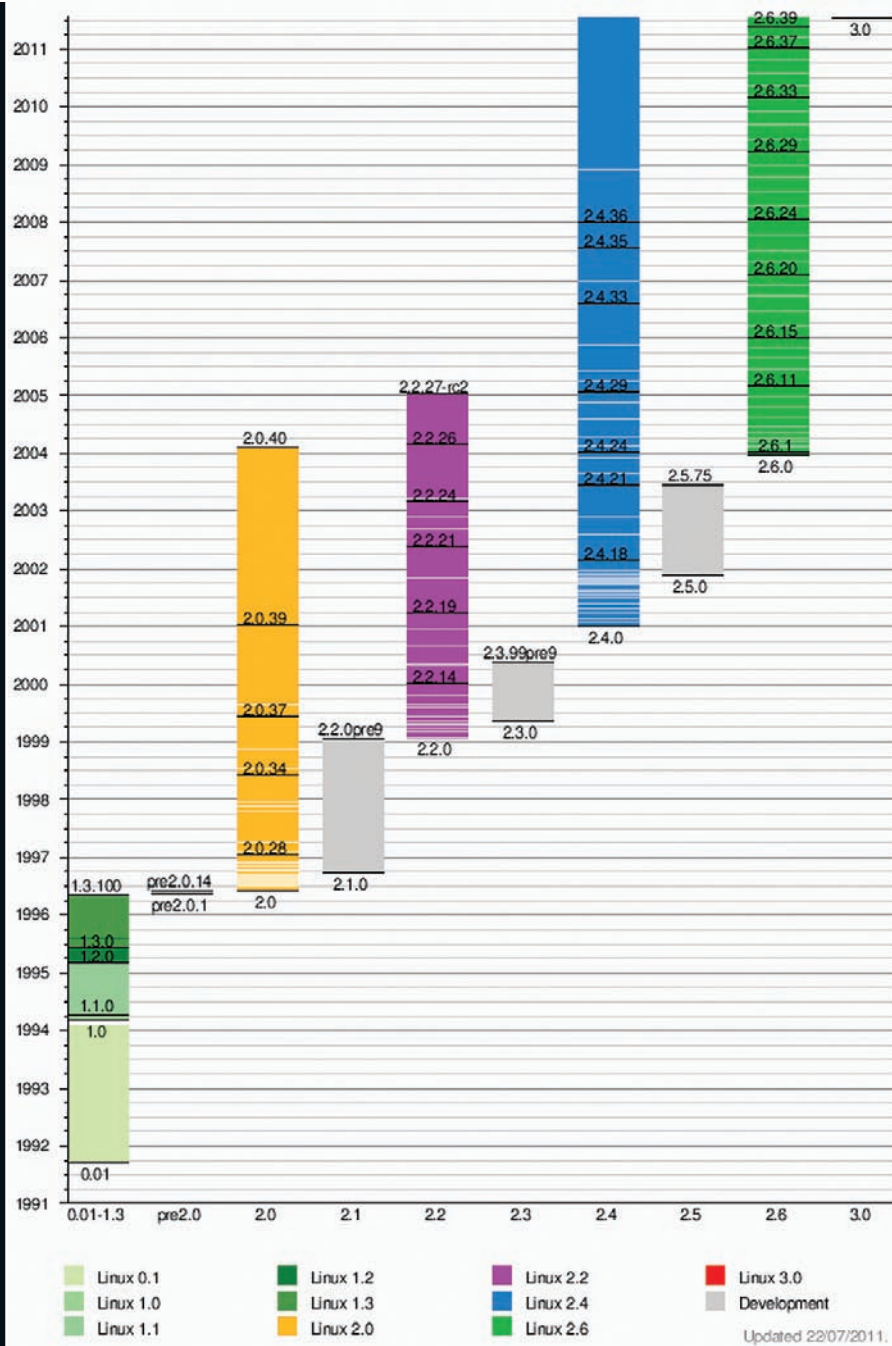
Взаимодействие между гостевой ОС и главным компьютером или другой гостевой системой осуществляется через сетевое соединение, и виртуальные машины традиционно эмулируют настоящую сетевую карту для более широкой совместимости. В Linux недавно появился сетевой драйвер *vhost*, ликвидировавший максимум устаревшей аппаратной эмуляции и достигший скоростей до 10 ГБ/с, а также меньшего времени задержки.

При этом он работает со стандартными драйверами в гостевой системе, а стало быть, в модификации виртуальной машины нет необходимости: вся работа выполняется в ядре на главном компьютере.

Подъем Android

Одна компания сумела резко увеличить число пользователей систем на базе Linux, и речь не о Sanoical. Операционная система Android от Google (буквально) вложила Linux в руки миллионам людей, которые до этого вообще не знали, что такое ОС, не говоря уже о Linux. Хорошо ли это? Время покажет, но тут есть что обсудить.

Многих также заботит участие (или неучастие) Google в разработке ядра. Многие сделанные в Google изменения не возвращаются в ядро. Лицензия GPL такое позволяет, раз уж доступен исходный код, но многие чувствуют, что это не в русле стремления поделиться, присущего GPL.



› Временная шкала развития ядра Linux показывает, что серия 2.6 была «самой свежей и передовой» добрых восемь лет. Переход к 3.0 явно приподнялся.

Зависит ли это от нежелания Google или так устроен их процесс разработки (похоже, они предпочитают возвращать код изредка и большими блоками, в противовес подходу открытого ПО «поямалу и часто»), тоже покажет время.

Однако для Linux это огромный, пусть и по большей части невидимый, шаг вперед за последние несколько лет. Эх, кабы Линус предусмотрел принятие лицензии, которая требовала бы изображать пингвина Тукса на любом устройстве с Linux... Кстати, Android – сильнейший аргумент, чтобы называть настольную/серверную ОС GNU/Linux, ради ее отличия от других систем на базе Linux, типа Android, и доказательства, что Linux без GNU вполне возможен и популярен.

Хотя за последнее время в ядре Linux не было явных больших скачков, это лишь признак его зрелости и оправдание ярлыка «3.0». Оно продолжит эволюцию, развиваясь (в основном) небольшими шагами, что может показаться заурядным. Но если вы когда-либо усомнитесь в прогрессе Linux за последние несколько лет, возьмите экземпляр *Linux Format* трехлетней давности, прочтите, что считалось новым и интересным тогда, а потом попробуйте установить дистрибутив с DVD на свой новый компьютер.

С нетерпением ждем новых функций в ядре 4.0, которое выйдет в 2021 году – даже если это будет всего лишь переименование 3.99. LXF

ВНУТРИ
Магеіа



В ЧЕМ МАГИЯ МАГЕІА.

Это самый первый релиз —
и мы уже подпали под его чары.
Исследует Маянк Шарма.



огда было объявлено о проекте Mageia, немногие сомневались, что он выйдет, но почти всех удивил выпуск полнофункционального дистрибутива менее чем через год.

И за девять месяцев после ответвления от Mandriva проект Mageia достиг намного большего: он сумел выполнить свое обещание и собрал – с нуля! – по-настоящему меритократичный проект, опирающийся на сообщество.

Сообщество Mageia выросло с десятка до более чем 100 разработчиков, которые работают примерно в 12 командах. Сообща они создали более 14 000 двоичных пакетов, поддержку разрядности как 32, так и 64, и уже работают над архитектурой ARM.

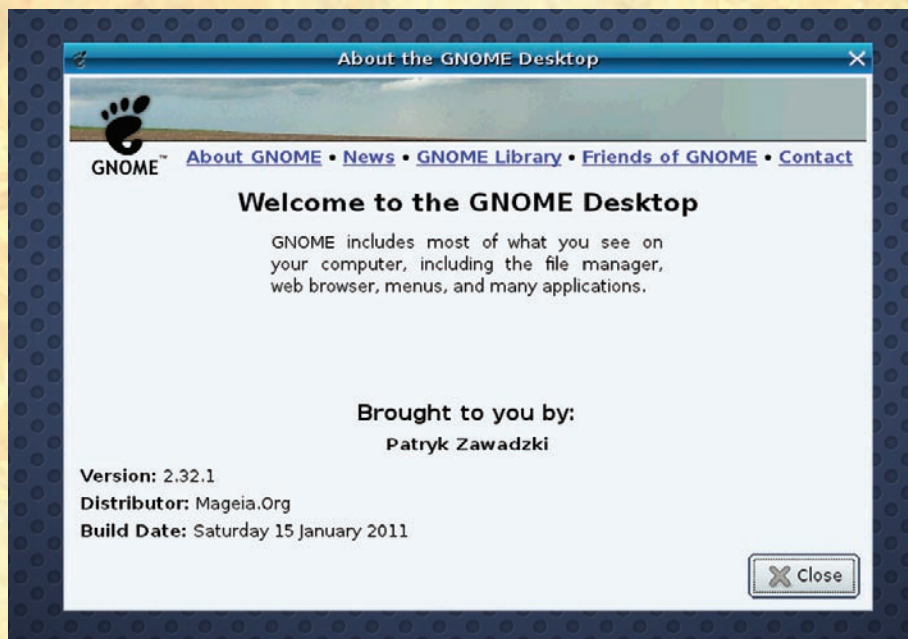
Первому релизу аплодирует почти все сообщество открытого ПО. Члены проекта скромно объясняют свой успех «опорой на плечи гигантов» – для дистрибутива это, может, и правда, но она не учитывает усилий сообщества по обустройству инфраструктуры и подготовке экосистемы, чтобы подпитывать Mageia и способствовать вкладам.

Анонс ответвления

Еще в сентябре 2010 коллектив бывших сотрудников Mandriva и членов его разношерстного сообщества решил ответить от дистрибутива, после того, как родительская компания, Mandriva SA, ликвидировала одну из своих компаний и решила сместить интересы с глобального рынка в Южную Америку, откуда поступала большая часть дохода компании. Потеряв уверенность в будущем любимого дистрибутива, сообщество, вполне в стиле открытого ПО, решило взять дела в свои руки.

Анна Николая [Anne Nicolas], бывший технический директор Mandriva и член совета учредителей в Mageia.org, подчеркивает, что решение создать Mageia не было импульсивным.

«Прошлым летом мы много говорили с главными представителями сообщества, пытаюсь понять, есть ли у нас общие интересы в запуске нового проекта», сказала она. И добавила: «Исходная команда состояла из тех, у кого были мотивация и опыт довести проект до завершения,



➤ Лучшее в подходе Mageia к разработке, ведомой сообществом — прозрачность всех аспектов проекта.

четкое представление о том, что нам нужно, и хорошие причины сделать это».

Оливер Бургер [Oliver Burger], многолетний пользователь Mandriva и член германского сообщества пользователей Mandriva, MandrivaUser.de, вошел в проект Mageia с самого начала.

Бургер объясняет, что за последние годы его и несколько других ребят в MandrivaUser.de сильно расстраивала манера Mandriva SA обращаться с сообществом.

«Нам действительно нравился Mandriva Linux, и мы не хотели менять дистрибутив, а Mageia – как раз то, чего мы ждали, – отметил он. – А когда я познакомился с другими членами проекта, я стал чувствовать себя здесь как дома».

Николя вспоминает, что после анонса настали часы нервно грызть ногти. «У нас не было ясного представления о настроениях в сообществе, – говорит она. – Но спустя несколько часов мы почувствовали себя более уверенно».

Одним из приоритетов проекта было контактировать с сообществом, держа его в курсе происхо-

дящего. Николая объясняет, что кроме применения почтовой рассылки, wiki-страницы и IRC для поддержания участников в процессе, команда приняла решение вести регулярный блог.

В одном из самых первых постов блога после объявления об ответвлении Николая написала, что положительная реакция укрепила их решение.

«Глядя, как подтягиваются люди из сообщества Cooker и пользовательского сообщества, а также те, кто ранее нас покинул, мы утвердились в том, что решение создать дистрибутив, полностью разрабатываемый сообществом, было и остается правильным».

Прозрачность – один из отличительных признаков дистрибутива, и это также распространяется на связи между разработчиками. Первоначально команда обходилась почтовыми рассылками



➤ Мы высоко оценили Mageia и ставим ее наравне с солидными дистрибутивами.

Заварим кашу?

Версия разработчиков следующего дистрибутива Mandriva называется Cooker [англ. кухонная плита]. Версия Mageia от разработчиков, Cauldron [англ. котелок], аналогична по масштабам и целям, и, подобно Cooker, является скользким релизом.

Новые пакеты добавляются и обрабатываются в Cauldron. Ближе к концу цикла нового стабильного дистрибутива репозитории замораживаются вплоть до выхода нового дистрибутива, и тогда работа над Cauldron возобновляется. Cauldron не запускается как обычный дистрибутив, он даже может быть не устанавливаемым.

Это также не то место, где можно добыть свежие программы для стабильного релиза; помимо своих собственных капризов, он может оказаться скомпилированным с библиотеками, которые у вас отсутствуют.

Cauldron составлен из нестабильных пакетов, которые ломаются неожиданно, до того, как их исправили разработчики.

Кроме составителей пакетов, Cauldron полезен только для тех, кто хочет протестировать релиз разработчиков, чтобы убрать шероховатости в следующем стабильном релизе.

и IRC, и общение было слегка неформальным, но Николая говорит, что очень скоро были введены еженедельные собрания, чтобы люди видели, что происходит, и могли принимать решения по общей стратегии.

Бергер рассказал, что изначально в почтовой рассылке дискуссии шли в основном на технические, организационные и юридические темы, но в процессе формирования команды, который начался в декабре 2010, обсуждения переместились в другие области.

«Например, в команде интернационализации нам пришлось выбирать утилиты, – говорит Бергер. – Через некоторое время мы остановились на *Transifex*, поскольку она уже широко использовалась и можно было расспросить людей из Fedora и других мест об их опыте работы с ней».

Николай заключает: «Сейчас можно сказать, что общение проходит хорошо. Одна из важных целей состоит в поддержании горизонтальных связей внутри Mageia – например, чтобы технические и нетехнические команды регулярно общались друг с другом».

Хорошее управление

С ростом команды проекту потребовался более удобный способ для взаимодействия ее членов, так что была выработана структура управления, разделившая проект Mageia на четыре сегмента – сообщество в целом, команда, общественный совет и правление.

Правление отстаивает ценности проекта и руководит ассоциацией Mageia.org. Совет выполняет ежедневное управление, планирование, координацию и производство. Есть также команды, ответственные за другие области дистрибутива, такие как формирование

и сортировка пакетов, QA, документация и маркетинг.

Определив роли субъектов в модели управления, первоначальная команда может делегировать задачи влившимися



Многие утилиты Mageia отличаются от своих аналогов в Mandriva лишь легкой косметической отделкой.

Поддержка обновлений

Пользователей Mageia 1 могло озадачить первичное отсутствие обновления пакетов. Это оттого, что разработчики были недовольны обработкой обновлений в Mandriva, и политика обновлений в Mageia все еще в работе.

«Мы были не вполне довольны старым процессом, и нам нужно было, чтобы новое сообщество пользователей и пакетчиков повлияло на способ его осуществления», записал в блоге Стью Бенедикт [Stew Benedict] из команды безопасности.

«Так, после обсуждения и небольшой закусочной работы над механизмами применения обновления у нас создан процесс, когда команда безопасности, команда тестировщиков и куратор пакетов совместно работают над сборкой, тестированием и применением обновлений», продолжает Бенедикт.

Во время переходного периода с Mageia 1 на Mageia 2 команда делает исключение для политики обновлений – это касается пакетов, недоступных в Mageia, но доступных в Mandriva.

Эти пакеты будут рассматриваться как обновление, если они сумеют пройти процесс тестирования Mageia. После релиза Mageia 2 недостающие пакеты проведут через репозитории обратной поддержки, которые впоследствии будут использоваться для новых релизов пакетов.



Графическое управление пакетов Mageia 1 идет на одном дыхании.

участникам, не опасаясь, что дела могут пойти не по плану.

Проект также занялся работой по созданию и регистрации ассоциации Mageia.org.

«Это было сделано, чтобы Mageia могла получать финансирование, а также для хостинга помощи и многих других вопросов, – объясняет Николай. – Мы хотели, чтобы Mageia стартовала в хороших условиях, и хостинг был одним из главных забот и приоритетов».

Ассоциация Mageia.org была зарегистрирована 7 октября 2010 года. На вопрос о модели прибыли, заданный во время интервью на Open World Forum в Париже в том же месяце, Николая ответила, что вокруг Mageia будет сформирован целый ряд услуг, включая поддержку, консалтинг и разработку.

Первая встреча Генеральной Ассамблеи новой организации прошла в феврале 2011 в кулуарах FOSDEM. Николая избрали президентом, Михаэля Шерера [Michael Scherer], давнишнего участника Mandriva – секретарем, а Дамьена Лальмана [Damien Lallemand], бывшего работника Mandriva и менеджера по качеству дистрибутива, избрали казначеем. На встрече Лальман представил подробности финансового состояния проекта. Было получено €8833 пожертвований, а потрачено €3310, в основном на серверы и другое оборудование.

22 октября 2010 года заработали серверы Mageia. Чтобы обеспечить вклад сообщества в развитие проекта, команда приняла решение раздобыть хостинг и создать систему сборки. Это была грандиозная задача. Изначально инфраструктура основывалась на Mandriva Linux One. Для задач администрирования применялась платформа для управления системой Puppet, подогнанная к Mageia. Команда также установила скрипт Lurt Perl, чтобы перекомпилировать SRPM. Lurt отсле-

живает список пакетов с разными архитектурами и перекомпилирует каждый в окружении chroot.

Другой важной задачей, кроме установки системы сборки, было развертывание централизованного решения аутентификации с использованием LDAP, чтобы получить web-интерфейс, дающий пользователям и участникам возможность зарегистрироваться в любой службе Mageia, например, в Bugzilla, SVN, списках рассылки и на форумах.

К моменту, когда команда дошла до установки subversion и принятия вклада участников, они уже отставали от графика. Оглядываясь назад, Николай говорит, что можно было сделать все лучше – например, установить временную систему SVN, куда участники могли бы вносить изменения.

Начальная раскрутка

Пока устанавливалась система сборки, исходная команда Mageia также взялась создавать из разработчиков, которые зарегистрировались на вики-странице Mageia, ориентированные на решение задач группы.

Некоторые нетехнические команды – дизайн, маркетинг и так далее – были уже сформированы. Вскоре появились команды пакетчиков, сортировщиков ошибок, тестеров и переводчиков, и лидер каждой из них был представлен на общественном совете Mageia.

Бургер, который является главой команды интернационализации, также тесно работает с сист. админом. Он считает, что формирование команд и инфраструктуры было нескладным, но усилия оправдали себя.

Но дело еще не было завершено! Следующим шагом было сборка начальной загрузки, небольшого инсталлятора, используемого для загрузки большей системы. В блоге Николая объяснила, что в случае с Mageia начальная загрузка – это то ме-

сто, где собираются базовая система и основные зависимости, так что все приложения можно там собрать, используя собственные средства и пакеты Mageia. Это стандартная практика для всех дистрибутивов Linux.

Но что произойдет с такими пакетами, как *GCC*, *Glibc* и прочее, которым для компиляции требуются они сами? В Mageia эту проблему курицы и яйца решили, позаимствовав низкоуровневые пакеты у Mandriva.

Однако, прежде чем их использовать, их пришлось отчистить от всего связанного с Mandriva – например, изображений копирайта, значков и RPM-макросов.

Правильная начальная загрузка имеет решающее значение для любого дистрибутива, не только для ответвления. Для нового дистрибутива написание и документирование начальной загрузки пригодится при добавлении поддержки новой архитектуры.

Создание начальной загрузки и сборка пакетов для нее в Mageia отняло месяц, что привело к задержке релиза ISO изначальной альфа-версии на пару месяцев. Но как только все улеглось и встало на свои места, проект стал укладываться во все последующие сроки релизов.

Свет мой, зеркальце

Другой важный шаг в новом дистрибутиве – определить структуру репозитивов. Прежде чем Mageia смогла принять решение об этом, команде пришлось выбирать приемлемые лицензии, что влияет на программы, включаемые в дистрибутив.

После долгой дискуссии было решено распределить все программы по четырем репозиториям.

«База [core]», активный по умолчанию, содержит только те программы, чьи лицензии одобрены FSF или OSI.

«Несвободный [non-free]» репозиторий, ныне задействованный по умолчанию в Mageia 1, содержит свободно распространяемые бесплатные программы и прошивки, исходный код которых недоступен: например, проприетарные графические драйверы от ATI и Nvidia.

Третий, «грязный [tainted]» репозиторий по умолчанию выключен. Он содержит ПО, спо-

собное вызвать патентные проблемы, ограничивающие распространение, например *VLC*, *K9Copy*, *LAME Encoder* и так далее.

Репозиторий «отладки [debug]» содержит отладочные RPM для программ из остальных репозитивов.

Наряду с пакетами и политикой обслуживания зеркал, проект был готов принять зеркала. К ноябрю 2010-го команда заказала два зеркала, чтобы синхронизировать Mageia.

Николя объясняет, что после первого ISO разработчиков зеркал у проекта добавилось.

«Люди обрели уверенность в Mageia как в реальном проекте», прибавила она.

Финальный отсчет

По завершении основной работы были распахнуты ворота в сообщество Mageia. Команда завершила логотип проекта в декабре 2010-го, перепробовав за месяц более 430 вариантов дизайна.

После этого инфраструктура была сформирована, команды – организованы, система сборки – установлена, а пакеты начальной загрузки – вычищены. Альфа 1 версия Mageia 1 вышла 15 февраля 2011 года, с двухмесячным опозданием.

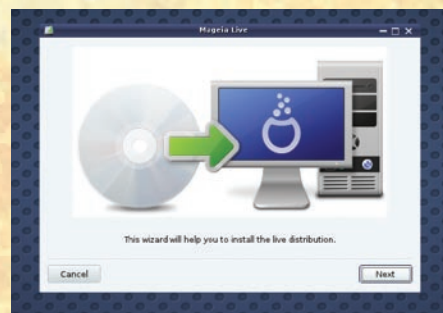
Для разработчиков и участников релиз дал «незыблемый» фундамент, на котором закладывалась последующая работа.

В записи блога, где объяснялась важность этого, Николя сообщила: «Питание подано, фабрика работает, команды организованы, и все это позволяет вам представить минимальный работающий продукт, в его первоначальном виде».

За этим последовал первый день тестирования Mageia, 26 февраля 2011 года.

Система сборки Mageia уже работала «на полную мощность», вырабатывая для репозитивов более 6100 пакетов, содержавших *LibreOffice*, *Eclipse*, *KDE 4.6.1* и так далее.

Альфа 2 вышла 15 марта 2011 года, как в 32-, так и в 64-разрядном вариантах. Бета 1 была выпущена 5 апреля, там была проделана огромная работа по локализации. К зеркалам добавились торренты. Команда планировала выпустить live CD, однако столкнулась с трудностями сборки такой системы. Днями тестирования релиза стали 11 и 12 апреля.



► Кроме установочных и live-носителей, можно обновить установку Mandriva 2010.2 до Mageia 1.

26 апреля Mageia выпустила бета-версию 2, исправившую множество ошибок и добавившую много приложений, таких как *Firefox 4.0*, *Chromium*, ядро 2.6.38.4, *LibreOffice 3.3.2.2* и *KDE 4.6.2*. Пакеты были заморожены, и разработчики устремили силы на исправление ошибок и улучшения пользовательской среды. Этот релиз также имел live CD, один с Gnome, а другой – с KDE.

Единственный и неповторимый релиз-кандидат вышел 17 мая, с множеством исправлений и с дополнительными переводами.

Проект попросил пользователей проверить все проблемы, связанные с обновлением Mandriva 2010.1 и 2010.2, и прислать отклики по процессу установки.

И, наконец, после девяти месяцев интенсивной работы 1 июня вышел релиз Mageia 1, включающий все, что полагается иметь мейнстримовому дистрибутиву. Он содержал ядро 2.6.38.7, *KDE SC 4.6.3*, *Gnome 2.32*, *Xfce 4.8.1*, *Firefox 4.0.1*, браузер *Chromium 11.0.696.65*, *LibreOffice 3.3.2.2* и многое другое. Релиз был доступен как установочный DVD и как CD на две архитектуры (с ограниченным набором программ), а установщиком был традиционный DrakX. Есть также установочные live CD с KDE и с Gnome.

А пока дистрибутив карабкается на вершину чартов Linux и сообщество аплодирует его достижениям, разработчики берутся за активную работу над Mageia 2. **LXF**



Магия продолжается...

В интервью журналу *ToDo Linux* Михаэль Шерер объяснил, что замыслом первого релиза было убедиться, что у проекта есть солидная устойчивая инфраструктура.

«В сущности, мы хотели сделать первый релиз не для глубоких изменений нашей основы, а чтобы убедиться в устойчивости нашей организации», сказал он.

Так что же есть у Mageia для нас в перспективе? После довольно долгих дебатов был утвержден девятимесячный цикл релизов, и каждый релиз Mageia будет поддерживаться 18 месяцев.

Если после следующего релиза у проекта останется достаточно ресурсов, чтобы выполнять это обещание, будет рассмотрена возможность выпуска долгосрочной версии, с трехлетним сроком поддержки, каждые 18 месяцев.

Разработка Mageia 2 уже в пути; предложены более 70 изменений в разных частях дистрибутива, от начального установщика до базовой системы и инфраструктуры управления пакетами. Выход версии Альфа 1 у Mageia 2 запланирован на 16 ноября 2011 года, а окончательного релиза – 4 апреля 2012 года.



Desktop Summit 2011: Будущее Gnome и KDE

Ведущие игроки настольного Linux встречаются, чтобы обсудить свои партии.

Что получится, если свести воедино Gnome и KDE? Отличное ПО? Вечная война? Уже восемь лет две главных силы настольного Linux проводят ежегодные собрания, намечая планы на будущее. Первый раз эта встреча произошла в 2009, когда оба сообщества решили совместить проведение своих конференций. Два года спустя мероприятия объединили.

Более семи дней свыше 700 энтузиастов свободного ПО слушали около 170 докладов с последующим подробным обсуждением всех новинок в мире Gnome и KDE. Что ждет Gnome? Сможет ли KDE оторваться от настольных компьютеров? И как сделать тостер с нуля?

Gnome все еще ликует по поводу выхода третьей версии и многочисленных положительных отзывов о новом интерфейсе, Gnome Shell, попутно отвечая на критику со стороны тех пользователей, которым изменения не пришлись по вкусу.

В целом, публика оказалась довольна Gnome 3, но разработчики признали, что в нем есть и весьма противоречивые решения, и работа еще предстоит. Карен Сэндлер [Karen Sandler], исполнительный директор Gnome Foundation, повезло присоединиться к Gnome одновременно с выходом нового красивого рабочего стола для «нормальных людей».

KDE уже по большей части отделался от критики KDE 4.0 и вводит поэтапные изменения в рабочий стол, параллельно открывая новые горизонты, такие как планшеты. Сквозной темой выступлений KDE на саммите было расширение круга устройств. Но-

вый интерфейс KDE для планшетов, Plasma Active, основан на нескольких существующих технологиях данного проекта.

Себастиан Кюглер [Sebastian Kügler] объяснил, что пользовательский интерфейс

ориентирован на род работы и подстраивает оформление и так называемые «Комнаты [Activity]» под конкретные действия пользователя.

«Gnome ликует по поводу хороших отзывов о новом интерфейсе.»

Развивая эту тему, Марко Мартин [Marco Martin] описал, каким образом Plasma Active пытается понять привычки пользователя, используя «контекст, модели и действия». Контекст включает информацию о расположении, времени, активных файлах и приложениях, а также недавних действиях. Модели пользования хранятся и сопоставляются с текущим контекстом, после чего пользователю предоставляется выбор ранее определенных Комнат в порядке релевантности. К примеру, система могла отследить, что пользователь просматривает сайты о Риме, и предложить переключиться на сохраненный вариант Комнаты, связанной с запланированным отпуском в Италии.

Plasma Active все еще находится в активной разработке, но для нее доступны регулярно обновляемые тестовые образы на базе OpenSUSE и MeeGo (см. community.kde.org/Plasma/Active)

Контекстуальное поведение Plasma Active основано на использовании Комнат и Перомик, впервые появившихся в KDE 4.0. Иван Чукич [Ivan Čukić] объяснил, как их взаимодействие помогает тем, кто работает сразу над несколькими проектами. Он привел в пример архитектора, который копирует на рабочий стол файлы, относящиеся к конкретному проекту, для быстрого доступа к ним, а когда работа закончена, возвращает их обратно в структуру каталогов клиентов и проектов.

Используя Комнаты, то же самое можно сделать проще. Plasma отображает любые файлы, независимо от того где они хранятся, прямо на рабочем столе, позволяя создавать множество Комнат и менять набор этих файлов по необходимости.

Управление свансами дает пользователю возможность держать окна открытыми в одной Комнате и скрытыми при работе в другой. Чукич отметил, что в работе над этим свойством Plasma Active они стоят лишь в начале пути. Сейчас Plasma сконцентрирована на окнах и приложениях, но в перспективе ориентация на документы, чтобы в Комнаты можно было попасть с разных устройств и необходимые файлы открывались в любом из доступных приложений.

Подключайтесь

Поскольку сегодня люди все чаще используют по несколько устройств, повысился интерес к удаленному хранению данных. Оуэн Тэйлор [Owen Taylor]

представил инновации в Gnome 3, упрощающие подключение к сервисам Google: учетные данные вводятся в одном месте, а затем определяется, какие сервисы синхронизировать.

Таким образом, одна учетная запись дает доступ к электронной почте, календарям и контактам, которые затем появляются

«В KDE внедряли свободную альтернативу web-сервисам.»



► Кнут Ирвин из Nokia оценил вклад таких сообществ, как KDE, в развитие Qt.

Настольный союз

Gnome и KDE проводят ежегодные европейские саммиты уже много лет. KDE Akademy появилась в 2004 г. (или 2003, если считать ее предшественника, меньшую по масштабам конференцию Kastle). Gnome GUADEC (Gnome Users and Developers European Conference) возникла еще раньше, в 2000 г. За эти годы ее принимали Бельгия, Чехия, Дания, Финляндия, Франция, Германия, Ирландия, Нидерланды, Норвегия, Испания, Турция и Великобритания.

В 2009, Gnome и KDE впервые совместили свои конференции на Gran Canaria Desktop Summit. В 2010 они опять разделились, но в этом году объединились вновь

в Берлине на Desktop Summit. На сей раз мероприятие стало не только совмещенным, но и общим, со смешанными секциями Gnome и KDE. Сообщества сошлись на том, чтобы проводить объединенные конференции раз в 2 года, поэтому следующий Desktop Summit пройдет в 2013 г.

В этом году конференция продолжалась три дня и завершилась традиционной ежегодной встречей каждого сообщества, а затем были еще три дня программирования, тренингов и встреч. Собранный более 700 участников Desktop Summit стал одним из самых больших событий для европейского свободного ПО.

в Evolution, клиенте групповой работы Gnome. А поскольку синхронизация выполняется при помощи демона Evolution Data Server, незачем держать Evolution постоянно открытым.

KDE также поработал над объединением web-сервисов, опробовав доступ к сообщениям из Facebook через групповой клиент KDE, Kontakt. В планшетном интерфейсе KDE, Plasma Active, реализована глубокая интеграция с «поделиться, понравилось, подключить», позволяющая быстро загружать контент, оценивать и комментировать чужие записи и вести обсуждения прямо с рабочего стола.

В KDE также внедряли свободную альтернативу проприетарным web-сервисам, через проект OwnCloud.

Проект предоставляет web-приложение для хранения документов, вкпе с интеграцией рабочего стола и web-интерфейсом. Основатель проекта Франк Карличек [Frank Karlitschek]

представил описание существующей версии, а также грядущей версии 2.0. OwnCloud 1.2 имеет web-интерфейс для доступа к файлам и дает возможность локального монтирования каталогов че-

рез WebDAV. Он также включает простой медиа-плеер в web-интерфейсе, предоставляя удаленный доступ к медиа-файлам. Версия 2.0 ожидается через несколько месяцев, с полностью обновленным web-интерфейсом и продвинутым медиа-плеером.

Шикарные дизайны

Общей темой для обоих сообществ стало привлечение дизайнеров – и улучшение дизайна – в свободном ПО. Этим вопросам были посвящены основные презентации Ника Ричардса [Nick Richards] из Intel и Клэр Роулэнд [Claire Rowland] из Fjord Design.

Ник Ричардс, плотно занимавшийся дизайном Gnome Shell, объяснил, что «индивидуальные разработчики погоды не делают», а тщательно продуманные правила дизайна могут помочь команде в создании, функционального и красивого ПО.

Клэр Роулэнд, исследователь способов использования компьютеров, возразила, что сегодня большинство устройств, которыми мы пользуемся ежедневно, нуждаются в однородном и интуитивно понятном дизайне. Задача дизайнеров – интерфейс, приспособленный под выполнение задач, а не презентация возможностей конкретного устройства.

Для этого необходимо привлечь квалифицированных дизайнеров в проекты свободного ПО. От их поддержки за последние годы выиграли и Gnome, и KDE. Сложность составляло лишь одно: дизайнерам необходимо было донести до программистов то,

Отпрокидывающая барьеры

Берлин – город, разделенный после Второй мировой войны и вновь воссоединенный вскоре после рождения Linux – прекрасное место для встречи двух сообществ, у каждого из которых (по крайней мере, когда-то) был свой путь.

Хозяин конференции, Университет Гумбольдта, находится в старой восточной половине города, откуда можно пешком дойти до бывшей стены, пропускного пункта «КПП Чарли» и Бранденбургских ворот. Даже сегодня, когда местные жители и гости столицы перемещаются свободно, различие между западом и востоком все равно ощутимо – в архитектуре.

То же происходит и с настольным Linux: у KDE и Gnome разные приоритеты, что очевидно по создаваемому ими ПО. Рабочий стол KDE все также настраиваем и пред-

лагает виджеты, которые можно расположить по своему усмотрению.

Gnome Shell – прочный пользовательский интерфейс, который можно применять ровно одним способом; зато в нем все просто и находится на расстоянии протянутой руки.

Но за этим фасадом – как и в Берлине – скрывается немало областей, над которыми оба проекта могут работать вместе.

Оба используют новшества вроде PulseAudio, NetworkManager и, все чаще, Zeitgeist.

Большая часть участников была удовлетворена широким охватом тем на саммите. Они получили отличный шанс посмотреть на работу коллег из другого сообщества и пообщаться с ключевыми игроками.

как приложение должно выглядеть и работать, а те должны были попытаться воплотить это видение, создать код и затем вновь обратиться к дизайнерам, чтобы они оценили это воплощение. Из-за рассредоточенности команды, типичной для свободного ПО, и тем, и другим тесно взаимодействовать было сложно. Gnome и KDE попытались облегчить задачу создания интерфейса пользователя, чтобы дизайнеры работали над ним напрямую, не обладая специальными навыками программирования.

KDE основывается на свойствах *Qt Quick*, позволяя разрабатывать интерфейс на JavaScript, а Лукас Роча [Lucas Rocha] представил *Gjs*, дающий возможность писать на JavaScript настольные приложения Gnome.

Козимо Чекки [Cosimo Cecchi] из Gnome говорил о новых виджетах для CSS-тем в *GTK 3*. Теперь любой человек с опытом веб-дизайнера может создавать темы *GTK*, а для приложений не на *GTK* стало проще подобрать компоненты стиля, чтобы они гармонировали с Gnome.



Собрав более 700 участников, Desktop Summit стал одним из крупнейших событий для европейского свободного ПО.

Закладывая основы

KDE воспользовался Desktop Summit, чтобы официально объявить свои планы для *KDE Frameworks 5*, библиотеки для поддержки будущего KDE 5.x. Противопоставив это переходу с KDE 3.x на KDE 4.x, Аарон Сейго [Aaron Seigo] пояснил, что теперь изменения будут поэтапными. Планируется сначала сосредоточиться на *Frameworks*, а уже затем на приложениях и новых версиях рабочего стола, и первое время изменения в библиотеках будут незаметны для конечного пользователя.

Помимо внедрения и развития технологий грядущего *Qt 5*, KDE также будет вести серьезную работу по модульной реорганизации своих библиотек, в чем есть несколько преимуществ. Во-первых, это упрощает сторонним разработчикам использование технологий KDE в своих приложениях, без привязки ко всему стеку KDE. Это также позволит уменьшить размер приложений KDE для мобильных устройств и создать для них автономные установщики под Windows.

Хотя Gnome 3 вышел совсем недавно, его разработчики тоже смотрят в будущее. Оуэн Тейлор задался вопросом, нужно ли вообще разрабатывать *GTK 4*, или же его место займут технологии Gnome Shell, подобные *Clutter*.

Разработчики *GTK* Маттиас Клазен [Matthias Clasen] и Беньямин Отте [Benjamin Otte] привели подробные выкладки о будущем *GTK*, оставив открытым вопрос: стоит ли *GTK* становиться мультипрограммным инструментарием и заниматься взаимодействием *GTK* и *Clutter*.

Эммануэле Басси [Emmanuele Bassi] сосредоточил внимание на будущем *Clutter*, представив свое видение следующей версии в качестве одного из слов *GTK 4* и выдвинув альтернативу напрямую взаимодействию с *GTK* в виде *Cairo* или *X11* или, в будущем, *Wayland*.

Общая картина

Улучшение настольного Linux – дело похвальное, но в ситуации, когда его использует малый процент людей, стоит ли тратить на это силы? Гейс Хиллениус [Gijs Hillenius], редактор Open Source Observatory and Repository (OSOR), выступил с обсуждением уровня использования свободного ПО в Европе и привел отрезвляющие цифры. По его данным, менее 1% гражданских служащих Европы пользуются рабочими столами не на основе проприетарного ПО.

Есть и более приятные новости. Французская жандармерия, штат которой составляет 100 000 человек, стала самой большой европейской организацией, сделавшей выбор в пользу свободного ПО. К 2015 году она планирует перевести все 85 000 единиц своей техники на Linux. Небольшие успехи есть в административном и образовательном секторах.

Организации, подобные OSOR, могут помочь связать компании, уже перешедшие на свободное ПО, с теми, кто только обдумывает такой переход, с целью консультирования и обмена опытом.

В мире технологий свободное ПО уже имеет вес. Кнут Ирвин [Knut Yrvin] из Nokia рассказал о том, как KDE предоставил 46 000 строк кода для *Qt* в 2010, и что 30% разработчиков *Qt* узнали о нем именно из проектов Open Source, таких как KDE.

Кому принадлежит код?

Одной из сильных сторон лицензий свободного ПО, таких как GPL, является то, что разработчики взаимодействуют друг с другом в системе понятных правил и требований. Однако некоторые организации и компании, включая проект GNU, заставляют разработчиков подписывать соглашение либо о передаче авторских прав, либо о предоставлении компании дополнительных прав.

Canonical запустил Project Harmony, с целью стандартизировать эти соглашения и уменьшить окружающую их путаницу, но сама

эта попытка выглядела весьма двусмысленно среди противников назначения авторского права. Поэтому все ждали дискуссии на данную тему в рамках Desktop Summit, собравшей ключевые фигуры свободного ПО, чтобы обсудить плюсы и минусы данной проблемы. В президиум вошли Марк Шаттлворт [Mark Shuttleworth], основатель Ubuntu; Майкл Микс [Michael Meeks] из сообщества LibreOffice и известный разработчик Gnome; и Брэдли Кун [Bradley Kuhn] из Software Freedom Conservancy и Free Desktop Foundation. Модератором заседания выступила Карен Сэндлер, исполнительный директор Gnome Foundation.

Обсуждение оживилось с самого начала, когда Марк заявил, что «свобода – не предмет для споров, свобода – это абсолют», но авторское право необходимо компаниям, чтобы иметь конкурентные преимущества. Брэдли возразил, что «свободу всегда оспаривают», и ее нужно активно защищать.

Майкла заботило, что авторское право противоречит сути свободных лицензий и ведет к монополизации. Он также высказал мнение, что подобные соглашения ослабляют сообщества: «Если люди не чувствуют, что им принадлежит по крайней мере часть кода, преданность проекту у них, в лучшем случае, слабеет».

Марк оспорил это, говоря, что авторское право тут не вредит, поскольку «в плане доступа к коду никакой монополии нет».

На это ему немедленно возразил Брэдли, утверждая, что в большинстве соглашений об авторском праве право на код всегда желает закрепить за собой одна компания. По его мнению, практические проблемы, такие как необходимость перелицензирования кода, можно обойти, передав авторское право компании, нацеленной на поддержку свободного ПО, такой как Free Software Foundation.

Марк парировал, что разумнее «доверять организации, которой предназначен код», и отметил, что «моральное право остается за проектом», которому принадлежит код.

Компромиссов с обеих сторон не наблюдалось, но по крайней мере, состоялась открытая дискуссия, касающаяся сути авторского права, с обсуждением существующих позиций перед лицом аудитории самих разработчиков ПО, которые, возможно, прежде даже не задумывались над этим всерьез.

Дебаты завершились полевым призывом Марка Шаттлворта к аудитории признать договор об авторском праве, поскольку «отдавать – значит, быть щедрым; попробуйте – получите огромное удовлетворение».

А многие участники-то думали, что работая над ПО под свободной лицензией, они и без того проявляют щедрость...

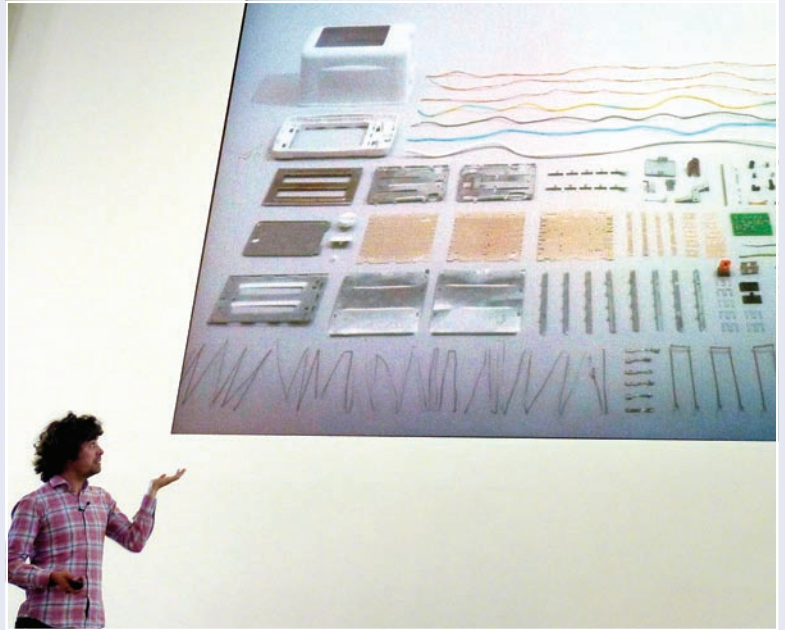
Как сделать тостер

Помимо разговоров о технических решениях и прогнозах в отношении свободного ПО, Desktop Summit радушно принял и Томаса Твейта [Thomas Thwaite], чей опыт в компьютерной сфере ограничился побегом из университета.

Представившись дизайнером и технологом, Томас рассказал, как решил сделать тостер с нуля, подумав, что устройство costing \$5 создать будет несложно. Он вознамерился обойтись без посторонней помощи, самостоятельно добыв себе железную руду и медь для электрических компонентов и никель из монет для нагревательного элемента.

В ходе своих экспериментов он выяснил, что в микроволновке можно плавить железо и что улитки падают до пластика, полученного из картофельного крахмала. Наконец Томас изготовил устройство, способное чуть-чуть подогреть хлеб, себестоимостью в £1200.

Его опыты (подробно описанные на thetoasterproject.org) не только повеселили публику и снискали похвалу технической



► Томас Твейт разобрал тостер, а затем сделал свой.

аудитории, но также донесли и серьезную мысль. Бытовые приспособления вроде тостеров кажутся не столь важными, но позволяют сэкономить время для более интересных вещей. Томас также отметил, что для достижения чего-то действительно важного надо работать в команде.

Unity: будущий союз?

За всеми обсуждениями Gnome и KDE осталось мало времени поговорить о Unity, попытке Canonical создать новый интерфейс для Linux; ныне это интерфейс по умолчанию в Ubuntu.

Большинство присутствовавших не волновалось о конкуренции, предлагая рассмотреть новые возможности и послушать, какие идеи есть у пользователей. Однако вопросы о Unity задавались на пресс-конференции: уделяли ли Gnome и KDE столько же внимания

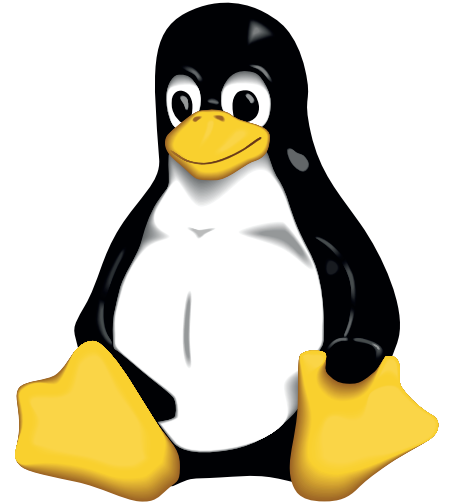
удобству использования, и не будет ли следующий Desktop Summit совместной конференцией Gnome, KDE и Unity. По первому вопросу Сторми Питерс [Stormy Peters], ранее исполнительный директор Gnome Foundation, а ныне глава разработчиков Mozilla, указала на существующие решения, такие как Gnome Usability Hackfest. Мирко Бём [Mirko Boehm], один из главных организаторов Desktop Summit, упомянул OpenUsability.org, осуществляющий обратную связь от пользователей к разработчикам.

Мирко высказал мнение, что как новый проект, Unity должен еще проявить себя, и нужно еще посмотреть, будет ли он столь актуален к следующему Desktop Summit. Дэйв Нири [Dave Neary] из Gnome предпочел не сводить все к Unity, сказав, что был бы рад, если бы грядущий Desktop Summit привлек к участию не только Gnome и KDE, но и разработчиков других ведущих проектов Linux, таких как LibreOffice.

Говоря о другом союзе, Аарон Сейго из KDE ответил на вопрос о взаимодействии на саммите обоих сообществ, отметив, что никого уже не удивляет сотрудничество Gnome и KDE, и эта работа ни в коем случае не ограничивается Desktop Summit. Команды поддерживают постоянный контакт в разработке совместных и обоюдно интересных технологий и в течение года даже практикуют взаимопосещения собраний разработчиков. KDE и Gnome делают разные вещи для разных аудиторий, но у них много общего, и вместе они сильнее. LXF

«Томас изготовил устройство, способное подогреть хлеб, за £1200.»

Linux: Обращение к пользователю



Исторические изыскания **Алексея Федорчука** добрались до крутого разворота Linux лицом к пользователю.

Red Hat, проторивший путь дистрибутивам для всех с коммерческой поддержкой, не долго оставался на нем одиноким: вскоре у него появились последователи. И первым среди них стал Suse.

Дистрибутив Suse изначально назывался S.u.S. E. – эту аббревиатуру производят от имени компании – его разработчика и распространителя (Gesellschaft für Software- und System-Entwicklung), хотя в источниках можно встретить и другие трактовки этого имени. Что, впрочем, не важно: форма его несколько раз менялась (S. U.S.E, SUSE, Suse), а в конце концов в чистом виде оно просто вышло из употребления.

Компания S.u.S. E. начинала свою деятельность с распространения уже имевшихся дистрибутивов – SLS и Slackware, а также издания руководств по UNIX и Linux. Однако вскоре ей захотелось иметь дистрибутив собственный, каковой и был создан на имеющейся кодовой базе названных систем. Позднее этот путь станет типичным для большинства коммерческих Linux-компаний.

Suse, как коммерческий продукт вышеозначенной фирмы, начал развиваться практически одновременно с Red Hat, в 1994 году. Сначала это был самый обычный дериват Slackware с германской локализацией и прикрученной универсальной системой установки и конфигурирования – YAST, призванной свести тяжкий труд администратора к простановке галочек в должных полях.

Таким образом, Suse стал первым дистрибутивом со сквозной системой установки и автоматизированного конфигурирования. При этом ручные настройки путем прямой правки конфигов – единственный способ доведения до ума материнской Slackware – отменялись напрочь: попытки ручного вмешательства в Suse влекли в те годы последствия непредсказуемые.

Постепенно Suse все больше отдалялась от родительской системы, заимствуя многие особенности Red Hat. В частности, в нее были внедрены формат пакетов RPM и система инициализации в стиле System V (в Slackware, как известно, исконным был BSD-стиль инициализации). И в результате уже через пару-тройку лет Suse стала настолько своеобразной системой, что и сам папа Патрик не догадался бы о ее происхождении.

Еще одной особенностью Suse стало повышенное внимание к графической системе, место которой в Linux к тому времени

практически безраздельно заняла свободная инкарнация Иксов – XFce86. Тесные контакты с разработчиками последней обеспечил дистрибутив поддержкой самых современных тогда видеочипов – в те времена для каждого графического чипа (или серии близких чипов) предназначался собственный X-сервер.

Свидетельствую как очевидец: запустить Иксы на считавшихся тогда крутыми видеокартах типа ATI Match32/64 в Slackware или Red Hat часто удавалось только после прикручивания к ним соответствующего X-сервера, вытасченного из Suse.

Бизнес-модель Suse строилась несколько по иной схеме, нежели у Red Hat. В частности, этот дистрибутив включил ряд закрытых проприетарных компонентов (в первую очередь – ту же систему YaST). И в «полноразмерном» виде бесплатно не распространялся: для свободного скачивания была доступна функционально ограниченная версия.

Она же распространялась первыми системами онлайн-торговли по цене носителя и доставки. А полноценный дистрибутив в коробочном исполнении продавался за нема-

лые по масштабам тех лет деньги – от ста долларов и выше. Однако он сопровождался объемной (и очень качественной) печатной документацией, которая сама по себе составляла львиную долю стоимости коробки, и атрибутикой – в частности, значком хамелеона, исполненным не хуже иной правительственной награды.

Видимо, непосредственная продажа дистрибутивов и составляла основную статью дохода фирмы. Хотя и от технической поддержки в корпоративных масштабах она тоже, насколько я знаю, не отказывалась.

В скором времени Suse стал дистрибутивом номер один не только в Германии, но и практически по всей Европе, заняв на этом континенте ту же нишу, что и Red Hat в Америке. И жил долго и счастливо, пока не был куплен фирмой Novell – но тогда это было делом далекого будущего.

Caldera: предыстория грядущего

Лавры Red Hat не давали покоя майнтайнерам и на американском континенте. И из всех, вожделевших их в то время, наибольшая удача выпала на долю дистрибутива Caldera – правда, удача, так сказать, посмертная: в виде памяти, и не очень лестной. Хотя

«Этот путь станет типичным для коммерческих Linux-компаний.»

как раз собственно дистрибутивом такая память и не заслужена. Для понимания этого нам опять придется вернуться назад и даже свернуть в сторону – в мир проприетаризма. А именно – к истории компании Novell, которая уже фигурировала в нашем рассказе в связи с освобождением FreeBSD.

Возникнув в 1979 году, компания эта на протяжении долгого времени с миром FOSS никак не пересекалась, разрабатывая и продвигая свою проприетарную сетевую систему NetWare. И продвинула ее настолько, что одно время NetWare и корпоративные сети воспринимались как близнецы-братья.

Но в начале 90-х годов прошлого века компанию, как некогда сиятельного Камильбека из «Повести о Ходже Насреддине», поразило хватательное рвение. Она в массовом масштабе скупала операционные системы (DR DOS – продвинутый аналог MS DOS), компоненты офисных пакетов – текстовый процессор *WordPerfect* и табличный процессор *QuattroPro*, приобрела права на распространение настольной СУБД *Borland Paradox*. В числе прочего ею были получены также права на исходный код и торговую марку UNIX (оставшиеся как бы бесхозными после разделения их прародителя – AT&T). На базе чего была создана система UnixWare; но это части других историй – былой и грядущей.

«Офисное» направление деятельности Novell особого успеха не имело, как и ее UNIX-бизнес. В результате один из основателей фирмы, Рэй Нурда [Ray Noorda], ушел в отставку, а все новоприобретения были распроданы в розницу: офисные пакеты – фирме Corel, где они составили пакет *Corel Office*, агония которого продолжается чуть ли не по сей день, *UnixWare* – SCO (то есть Santa Cruz Operations), дополнив ее собственную систему, получившую отныне имя *SCO OpenServer*.

Рэй же Нурда основал новую компанию – Caldera Systems, которая, в частности, приобрела DR DOS у Novell. Однако основным направлением ее деятельности стала Linux-дистрибуция.

Первоначально Caldera Linux представляла собой цельнотаченый Red Hat, дополненный некоторыми не вполне свободными компонентами, типа рабочего стола *Looking Glass* и средств интеграции с сетями Novell NetWare. Однако вскоре этот дистрибутив обзавелся собственным графическим инсталлятором (одним из лучших для своего времени) и вообще приобрел своеобразие, отмеченное новым собственным именем – Caldera OpenLinux.

Свободная версия Caldera OpenLinux представляла собой очень компактный, аккуратно укомплектованный дистрибутив, широкому использованию которого препятствовали только слабые средства интернационализации. А коммерческая версия включала немало проприетарных продуктов, вплоть до *Wabi* (средства эмуляции Windows от фирмы Sun), текстового процессора *WordPerfect* и векторной «рисовалки» *CorelDraw*, работавших в режиме эмуляции, офисного пакета *StarOffice* (в то время еще закрытого), и так далее.

Наконец, в первых годах нового тысячелетия Caldera становится одним из соучредителей альянса United Linux – наряду с Suse (тогда это была Европа), Turbolinux (Япония) и Conectiva (Бразилия); таким образом, альянс этот охватил чуть не все континенты (африканской Ubuntu тогда еще и в проекте не было). Но это уже тоже начало иной истории.

А пока резюмируем: Red Hat, Suse и Caldera образовали к концу 90-х годов «могучую кучку», создавшую предпосылки для движения FOSS-систем в корпоративный сектор.

Магия Linux

Итак, на протяжении 1994–1997 годов дистрибутивы Linux обрастали «дружелюбными к пользователю» инсталляторами, средствами сквозного конфигурирования и пакетного менеджмента, и включали пользовательские, в том числе офисные, приложения. Предпринимались и первые попытки интернационализации. Однако от конечного пользователя эти



› Рэй Нурда пытался превратить Novell во второй Microsoft, но не смог.

дистрибутивы оставались не менее далеки, чем декабристы с Герценом – от народа.

Впервые о Linux для конечного пользователя можно говорить, начиная с 1998 года, когда Гаэль Дюваль [Gaël Duval] создал дистрибутив Mandrake (ныне – Mandriva). Основной его идеей было объединение Linux'a и графической среды KDE. Разработчики Mandrake были первыми, кто решился на такой шаг, несмотря на тогдашнюю неясность лицензионной политики в отношении библиотеки *Qt*, на которой основывался KDE: пуристы от FSF и, вместе с ними, основные майнтейнеры дистрибутивов, включая Red Hat, полагали лицензии этих программ не соответствующими идеалам свободного программного обеспечения. И попросту игнорировали единственную в те времена по-настоящему интегрированную пользовательскую среду.

Собственно говоря, первая версия Mandrake представляла собой самый обычный Red Hat (от которого унаследовала номер версии – 5.1) и KDE, уже тогда имевший большой набор штатных пользовательских приложений.

Второй особенностью Mandrake была попытка поддержки «из коробки» экзотического в те годы «железа», начиная от звуковых карт и кончая телетюнерами и цифровыми камерами.

Третья изюминка Mandrake – интернационализация: пожалуй, это был первый дистрибутив, декларировавший поддержку всех языков мира (что в итоге послужило предпосылкой для появления «русского Линукса»). Все это надолго приковало к Mandrake сердца пользователей, не имеющих прямого отношения к разработке программ или администрированию сетевых служб.

Роль Mandrake в приобщении первой волны конечных пользователей к миру Linux (и Unix-подобных операционок вообще) переоценить трудно. В том числе и потому, что этот дистрибутив обеспечивал, более или менее удачно, возможность не только работать, но и развлекаться – не особо утруждая себя настройкой аппаратных средств. Не случайно Mandrake (а теперь и Mandriva) уверенно лидировал в многочисленных опросах на форумах о первом в жизни дистрибутиве – вплоть до появления Ubuntu.

Таким образом, Mandrake можно считать практически первым дистрибутивом, поставившим во главу угла интересы конечного пользователя. Под его влиянием шаги в этом направлении предприняли и ветераны дистростроения, такие как Red Hat

► Газль Дюваль впервые объединил Linux и KDE.



и Suse – они обзавелись красивыми графическими инсталляторами, предлагавшими predetermined варианты установки типа пользовательской мультимедийной станции, офисной машины и так далее.

Немалую роль в обращении Linux'a к конечному пользователю (сиречь офисному и домашнему, профессионально не связанному с IT) сыграла директриса развития офисного пакета

StarOffice. Созданный немецкой фирмой StarDivision первоначально для OS/2, он быстро был портирован на все существовавшие тогда платформы и операционки, претендовавшие, хотя бы чисто теоретически, на звание настольных – в том числе и на Linux. И хотя *StarOffice* не являлся тогда ни открытым, ни свободным продуктом, он был доступен, при определенных условиях, для бесплатного использования. А по своей функциональности, опять же со множеством оговорок, приближался (или стремился приблизиться) к *MS Office*, ставшему безраздельным властителем столов конторских служащих (*Lotus Office* и *WordPerfect Office* к концу тысячелетия уже отошли в область преданий).

Исторической правды ради нужно заметить, что *StarOffice* был не первым офисным пактом для Linux: этот титул по праву принадлежит пакету *Applix* одноименной, также немецкой, фирмы. Каковой, правда, обладал рядом недостатков – в частности, не способен был без хирургического вмешательства работать с кириллицей. Да к тому же не был ни открытым, ни свободным, ни даже бесплатным.

Были и другие попытки создания программ офисного назначения. Тут можно вспомнить и *Siag* – прототип офисного комплекта, состоящий из текстового процессора и электронной таблицы; и простой монофункциональный текстовый процессор *Ted*; и *Lux* – попытку облечь TeX в клерковский костюм с галстуком. Все они канули в Лету забвения – только *Abiword* сохранился в составе эвентуального *Gnome Office* (хотя изначально к Gnome никакого отношения не имел). Впрочем, и *Lux* продолжает свое развитие – но уже скорее в качестве программы верстки «на скорую руку», нежели универсального word-процессора.

В результате всех этих процессов – и популяризации самой системы, и появления пользовательских приложений для нее – на рубеже 1998–1999 годов в широких кругах околокомпьютерной общественности заговорили о появлении «Linux с человеческим лицом». Казалось бы, этот самый «человеченный» Linux имеет все шансы прочно окопаться на пользовательских десктопах, заменив в этом качестве Windows (о прочих пользовательских платформах, за исключением MacOS, к тому времени забыли).

Началось явление, вошедшее в историю ушедшего тысячелетия как Linux-бум. Оно, в свою очередь, вызвало к жизни новые дистрибутивы, уже прямо заявленные в качестве пользовательских десктопов – такие, как Corel Linux, распространявшийся по схеме коммерческого софта. Впрочем, ни народной любви, ни более-менее приемлемой популярности они не снискали: настоящему о коммерческих (или квазикommerческих) дистрибутивах заговорят только через несколько лет.

Назад, в будущее: Gentoo и другие

Однако скоро выявилась и обратная сторона любого user-ориентированного дистрибутива «для всех»: оказалось, что у каждого из их разработчиков были свои представления о том, что же нужно конечному пользователю для полного счастья.

Одни полагали, что счастье достижимо только в среде KDE, другие – что истинно счастливым юзера может сделать только идеологически правильный Gnome. Ну, а третьи решали вопрос кардинально и помещали в дистрибутив все, что только можно. И юзерофильные дистрибутивы стали пухнуть, как на дрожжах. К тому же графические инсталляторы этих систем, облегчая, с одной стороны, установку, с другой – навязывали пользователю predetermined свше наборы приложений. О назначении коих этот самый пользователь рисковал никогда не узнать – просто из-за их избытия. А количество user-ориентированных средств конфигурации стало, по меткому выражению Владимира Попова, превышать число конфигурируемых параметров.

И вот тут многие пользователи начали вспоминать, что они еще и администраторы собственных компьютерных систем – пусть даже в масштабе одного отдельно взятого десктопа. И, с легкой руки Клиффорда Вольфа [Clifford Wolf] (создателя дистрибутива Rocklinux), в обиход вошел термин «дистрибутив, дружественный к администратору». Началась эра популярности дистрибутивов Source Based.

На этой волне появились и упомянутый выше Rocklinux (исторически первый дистрибутив из этой серии), и LFS (Linux from Scratch) – набор рецептов по сборке собственной Linux-системы с нуля, созданный Герардом Беекмансом [Gerard Beekmans], и вариации на тему Sorcerer, и CRUX с Archlinux.

Однако наибольшую известность на этом поприще снискал Gentoo Дэниела Роббинса [Daniel Robbins]. Не в последнюю очередь – благодаря прекрасной документированности процесса установки, позволяющей, строго следуя директивным указаниям, собрать индивидуализированную, в том числе оптимизированную под наличное «железо», систему даже относительно малоопытному пользователю. А система портежей, родившаяся под идейным влиянием портов FreeBSD, позволяла очень гибко наращивать функциональность системы установленной.

Правда, мало для кого Gentoo оказался первым дистрибутивом – миссия его заключалась скорее в повышении общей квалификации пользователей, прошедших через Red Hat, Suse или Mandrake и разочарованных их дружелюбием, местами навязчивым до неприличия. И с точки зрения понимания устройства системы опыт пользователя Gentoo уступает, пожалуй, только сборщику LFS.

Ну, а роль самого Дэниела Роббинса в пропаганде Linux также переоценить трудно. Кроме своего дистрибутива, он стал и автором многочисленных ярких статей о самых разных аспектах устройства этой ОС – и о файловых ее системах, и о программных RAID-массивах, и о приемах работы в командной оболочке. Увы, весной 2005 года он поступил на службу классовому врагу: по окончании им аспирантуры ни в одной фирме, связанной с Linux и Open Source, не нашлось для него должности с достой-

«Заговорили о появлении "Linux с человеческим лицом".»

ной зарплатой. Тем не менее, в среде майнтейнеров родного дистрибутива он был подвергнут анафеме. Правда, скоро выяснилось, что «Карапетяны в неволе не размножаются», и Дэниел с Microsoft распрощался. А проект Gentoo по-прежнему живет и развивается.

На пути к гармонии

Однако и тут все оказалось не так гладко, как виделось поначалу. Будучи исходно типичным представителем «дистрибутивов для себя», Gentoo весьма мало подходил на роль системы общего пользования. Установка его (точнее, сборка из исходников) вполне могла длиться сутками, после чего требовалась еще и ручная доводка – правкой конфигов в текстовом редакторе. А выигрыш от оптимизации «под железо» постепенно оказывался все более иллюзорным, нивелируясь ростом общей вычислительной мощности компьютерного парка. И среди пользователей началась неясная сначала тоска по прошлому – простым решениям «из коробки», позволяющим развернуть систему и начать работать в ней за считанные часы, а уже потом, при необходимости и по возможности, шлифовать ее до немемorable совершенства (а какой линусоид к совершенству не стремится?).

И тут наступил час систем, которым на всем протяжении своего развития удавалось счастливо балансировать на зыбкой грани между дистрибутивами «для себя» и «для всех», между универсализмом и индивидуализмом, не склоняясь ни к «популизму» Mandrake со товарищи, ни к «кастовости» Slackware (а позднее и Gentoo).

И здесь в первую голову нужно назвать одного из отцов дистрибутирования – Debian. Разработка в 1999 году *apt* (*Advanced Packaging Tools*) – универсального набора инструментов для



› Дэниел Роббинс, создатель Gentoo и автор многочисленных статей о Linux.

управления пакетами – сделала его лидером в пакетном менеджменте. Не случайно *apt* был использован в дистрибутивах на основе RPM (впервые – в бразильской Conectiva, затем – в российском Альт Линукс). И более того, он послужил прототипом для универсальных систем управления RPM-пакетами, таких, как *Yum* (Red Hat/Fedora, ASPLinux) и *urpmi* (Mandrake).

И именно поэтому Debian дал в нашем тысячелетии наибольшее число активно развивающихся потомков. Впрочем, это уже совсем другая история. **LXF**

Т е х н о л о г и я с ч а с т ь я



SUNRADIO.RU

сетевое радио под ключ на базе Linux
новое будущее вашей компании

pr@sunradio.ru | www.sunradio.ru

Оптическое распознавание текста

Что написано пером, то хорошо бы прочесть компьютером.
Тимур Мубаракшин рассматривает приложения, которые это умеют.



Наш эксперт

Тимур Мубаракшин имеет более чем 10-летний опыт работы в сфере ИТ, но с OCR-системами в Linux столкнулся недавно, и теперь переводит свой бумажный архив в электронный.

Linux не страдает от отсутствия утилит для распознавания текстов. Среди них и легкая в использовании и минималистская программа – *Cuneiform-Qt*, вышедшая из-под рук программиста ALT Linux Team Андрея Черепанова; и консольная утилита *GOOCR*; и *Tesseract* – программа с длинной историей; и некоторые другие программы. Мы попробуем дать здесь их беглый обзор.

Cuneiform-Qt

Cuneiform-Qt имеет очень минималистский интерфейс – кому-то он может показаться даже скучным; однако это не так. У программы есть все для того, чтобы удовлетворить потребности обычного домашнего пользователя: она позволяет открыть изображение в популярных графических форматах и распознать текст в этом изображении. Распознанный текст можно сохранить в формате HTML, обычном тексте или RTF. Перед распознаванием можно выбрать язык текста и формат. Пока поддерживается лишь работа с изображениями – получить текст прямо со сканера на данный момент нельзя.

Программа написана на C++ с использованием библиотеки *Qt4* под Linux; последняя стабильная версия на сегодня – 0.1.1. Выпускается *Cuneiform-Qt* под лицензией GPLv3.

На слабых компьютерах (768 МБ ОЗУ, процессор Intel Celeron 2400 Мгц) *Cuneiform-Qt* работает медленнее, чем хотелось бы, но это, видимо, особенности движка от *Cuneiform*, а не недостаток оболочки Андрея Черепанова. Иногда русский текст бывает нечитаемым, что, в свою очередь, можно списать на молодость версии продукта.

В целом *Cuneiform-Qt* производит положительное впечатление, и работает вполне сносно с небольшими объемами информации (до 10 страниц текста), так что ее можно посоветовать домашним пользователям.

GOOCR

GOOCR – мощная консольная утилита, способная работать со многими языками.

Разработку *GOOCR* начал в далеком 2000 году Йорг Шуленбургом [Jörg Schulenburg], и в дальнейшем у проекта появилась небольшая собственная команда. Разработка *GOOCR* идет не особенно шустро, однако основная функциональность для приложений подобного рода имеется. Для *GOOCR* существует удобная графическая оболочка *OCRFeeder*, о которой речь пойдет чуть ниже.

Чтобы внести ясность, упомянем, что *GOOCR* имеет второе

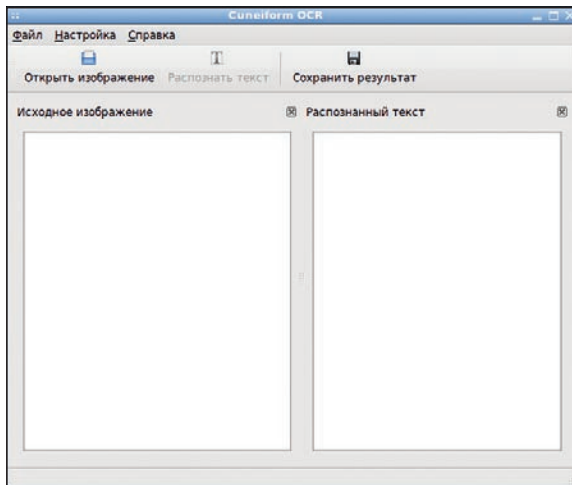
название – *JOOCR*: по словам автора приложения, так получилось потому, что на проекте sourceforge.net адрес gocr.sourceforge.net был уже занят. Но название *JOOCR* (Jörg's Optical Character Recognition – «OCR от Йорга») сейчас используется только для адреса страницы, хотя это и вносит некоторую путаницу.

GOOCR полностью поддерживает импорт изображений в форматах PNM, PBM, PGM, PPM и TGA; PCX поддерживается частично. Другие форматы изображений, такие как PNG, JPG, TIFF, GIF, BMP и rnm.gz/pnm.bz2, автоматически преобразовываются (используются утилиты сжатия *netpbm-progs*, *gzip* и *bzip2*).

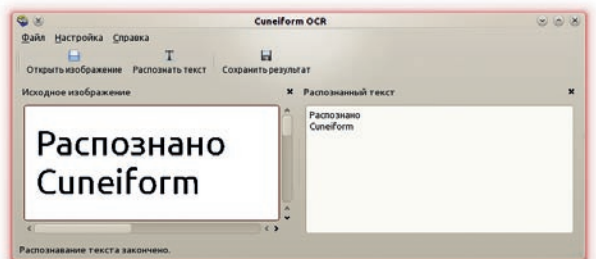
GOOCR поддерживает распознавание текста на основных европейских языках, однако русского, к сожалению, среди них нет. Поэтому *GOOCR* можно посоветовать, например, переводчикам. Процесс распознавания работает довольно быстро и не требует больших ресурсов компьютера.

Утилита может получать изображения напрямую со сканера (используя интерфейс *XSane*); распознанный текст может сохраняться в форматах TXT, ASC и HTML. Весь процесс распознавания будет виден в консоли.

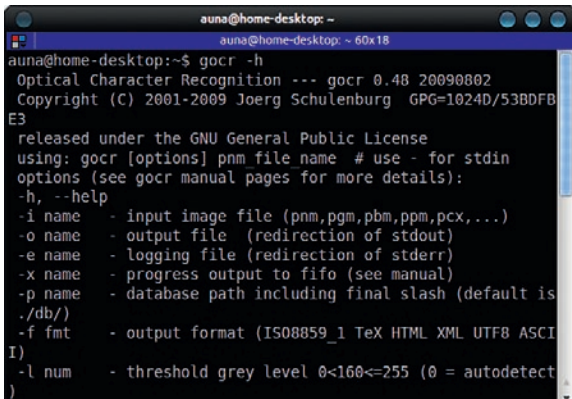
«Есть все, чтобы удовлетворить обычного пользователя.»



› Все самое необходимое в утилите присутствует, однако хотелось бы большего...



› Интерфейс *Cuneiform-Qt* совсем уж прост.



➤ **GOCR** может пригодиться тем, кому надо распознавать тесты на английском языке.

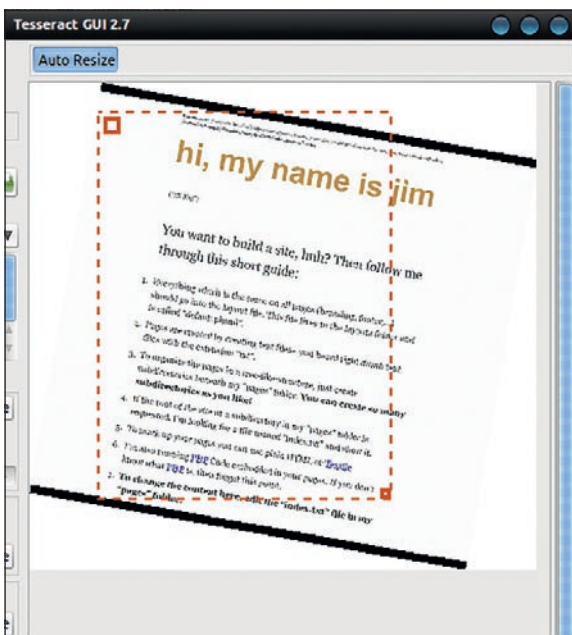
В качестве движка для оптического распознавания текста **GOCR** использует некоторые консольные и графические приложения, например **OCRFeeder**, и выпускается под лицензией GPLv2.

Tesseract

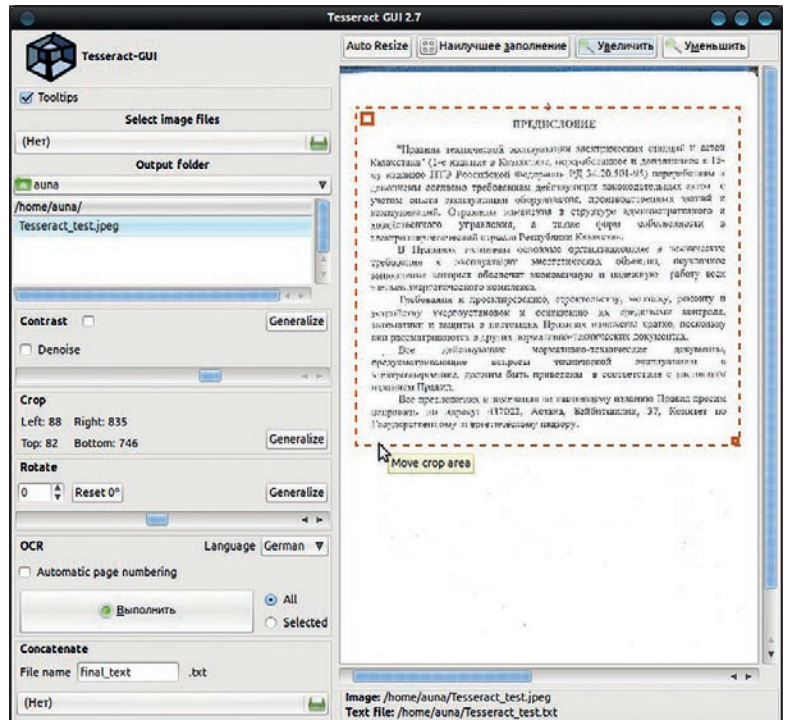
Tesseract – свободная программа для распознавания текстов, разрабатывавшаяся компанией Hewlett-Packard с середины 1980-х по середину 1990-х, а затем 10 лет «пролежавшая на полке». В августе 2006 г. компания Google купила ее и открыла исходные тексты под лицензией Apache 2.0 для продолжения разработки. В настоящий момент программа уже работает с кодировкой UTF-8; поддержка языков (с версии 3.0 – включая русский) осуществляется с помощью дополнительных модулей.

Ядро программы **Tesseract** разрабатывалось в Бристольской лаборатории Hewlett Packard (Великобритания) и в Hewlett Packard Co, в г. Грили, штат Колорадо, США, в 1985–1994 годах. В 1996 были введены значительные изменения и подготовлен порт для Windows. Значительная часть кода изначально написана на C, но проводились доработки для совместимости и с C++.

Для **Tesseract** имеется также графический интерфейс, который призван упростить использование программы для не искушенных в командной строке пользователей Linux – **Tesseract-GUI**, построенный на Python/GTK+.



➤ Возможности **Tesseract-GUI** довольно широки для простой оболочки.



➤ **Tesseract-GUI** облегчает доступ к консольным возможностям **Tesseract**.

Изображения, открываемые в **Tesseract-GUI** для распознавания, могут быть монохромными (черно-белыми), с градациями серого и цветными. Допускаются форматы изображений PNG или JPG, но для более качественного распознавания текста рекомендуется преобразовать изображение в формат хранения растровых изображений TIFF.

Tesseract-GUI дает возможность повысить контрастность изображения и изменить угол наклона (на исходном изображении изменения не отражаются). Распознаваться может как текст на всем изображении, так и текст выделенного участка изображения.

Для корректной работы оболочки требуется библиотека приложений, анализирующих и обрабатывающих изображения – **leptonica**. Установка **Tesseract** и **Tesseract-GUI** в большинстве популярных дистрибутивов происходит в штатном режиме с помощью соответствующего менеджера пакетов.

Tesseract-GUI сохраняет распознанный текст в файле формата TXT, в директории с распознаваемым изображением, однако можно указать свой каталог и дать имя файлу. Если в изображении две страницы (например, это разворот книги) то распознанный текст может быть сохранен как в одном, так и в нужном числе файлов постранично. Лицензия у **Tesseract-GUI** – GPLv2.

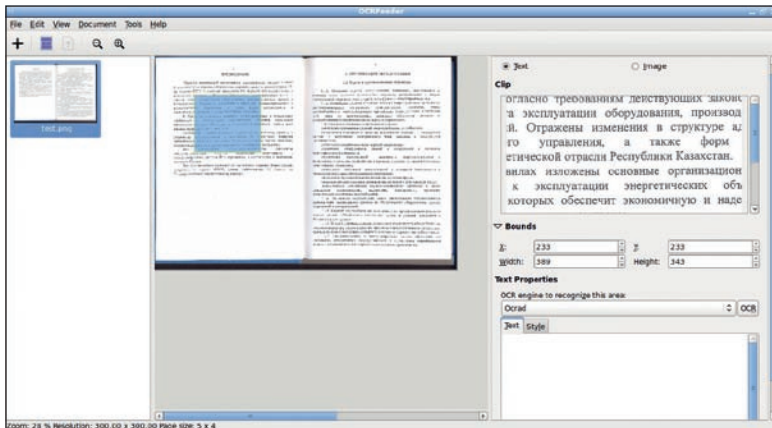
Tesseract-GUI работает очень быстро и подходит для большинства пользователей, которым требуется распознавание текстов объемом до 50 страниц. Поддержка русского языка не столь хороша, как, например, в **Cuneiform**, но все же имеется. Последняя стабильная версия приложения – 3.0.

OCRFeeder

OCRFeeder – удобный, написанный на Python и GTK+ графический интерфейс к нескольким консольным OCR-приложениям.

Разработка **OCRFeeder** начата как магистерская диссертация по специальности «Прикладная информатика» программистом Хоакимом Роча [Joaquim Rocha]. В дальнейшем приложение вошло в состав Gnome Project.

OCRFeeder автоматически определяет наличие установленных в системе OCR-приложений, а именно **Cuneiform**, **GOCR**, **Ocrad**, **Tesseract**, и использует их в качестве «движка», а для распознавания структуры документа применяется собственный алгоритм распознавания.



➤ **Плюсом OCRFeeder является его универсальность.**

Для поддержки распознавания языков в настройках всех перечисленных систем распознавания необходимо добавить аргумент, указывающий на этот язык. Например, для корректного распознавания текстов на русском необходимо добавить “-l rus”, а для правильной проверки русской орфографии пометить распознаваемый текст как русский.

Все настройки выполняются с помощью мыши.

В *OCRFeeder* можно импортировать данные как из графических файлов, во множестве популярных форматов (JPEG, PNG, BMP, TIFF, GIF, PNM, PPM, PBM и прочих), так и из файлов PDF. Также поддерживается захват изображения непосредственно со сканера. Изображения могут быть добавлены простым перетаскиванием в окно приложения из файлового менеджера или из контекстного меню *Nautilus*, файлового менеджера *Gnome*.

OCRFeeder позволяет очистить исходное изображение (без коррекции оригинала), в открытом изображении задать или изменить границы распознавания и выбрать наиболее подходящий для конкретного документа «движок» распознавания символов.

Имеется также возможность скорректировать нераспознанные символы, настроить стили параграфов, применить проверку орфографии в распознанном тексте с помощью библиотеки *libaspell* и выбрать шрифт для сохраняемых документов.

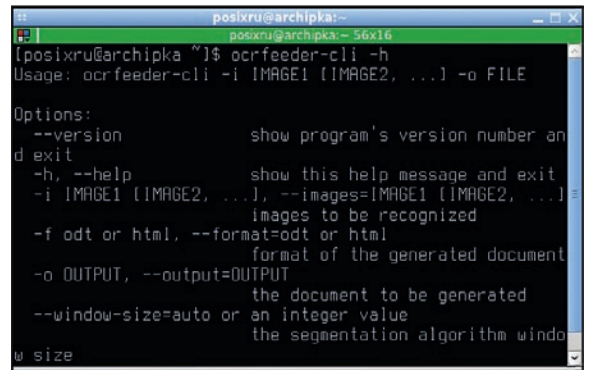
Основным форматом для сохранения результатов распознавания в *OCRFeeder* является ODT (OpenDocument Format). Текст также может быть сохранен в обычном текстовом формате TXT или в HTML.

Хотя *OCRFeeder* разрабатывается как приложение с графическим интерфейсом, имеется возможность работы и из командной строки с помощью утилиты *ocrfeeder-cli*. Это может оказаться полезным для автоматической пакетной обработки документов. Лицензия *OCRFeeder* – GPLv3. Последняя стабильная версия на сегодня – 0.7.5. Приложение можно посоветовать всем, кому нужно использовать несколько «движков» для распознавания тестов.

ABBYY FineReader for Linux

Нельзя обойти вниманием и коммерческую утилиту для распознавания текстов под Linux – уж слишком она известна.

«Ключи командной строки позволяют настроить параметры.»



➤ **Консольный интерфейс в OCRFeeder также присутствует. Правда, кому он нужен?**

FineReader – система оптического распознавания символов, разрабатываемая российской компанией ABBYY – слывет лучшей OCR-системой для Windows-машин. Версия для Linux не обладает графическим интерфейсом, что выглядит немного странно, поскольку за полную версию программы приходится выклады-

вать €149, и при этом количество распознаваемых страниц ограничивается величиной 12 000 в год. Имеются и более дорогие версии, в которых это количество значительно увеличено.

ABBYY FineReader for Linux – консольная утилита распознавания, пользующаяся набором подключаемых библиотек распознавания.

Приложение использует интеллектуальную OCR-систему и собственную технологию распознавания, позволяющую распознавать и воссоздавать документы со сложным форматированием.

С учетом немаленькой стоимости программы покупать ее для домашнего использования, возможно, и не стоит, однако для небольшой компании эта сумма выглядит вполне приемлемой.

Программа имеет множество ключей командной строки, которые позволяют гибко настроить параметры распознавания.

На сайте проекта имеется также и демо-версия, позволяющая распознать 100 страниц (после регистрации на сайте и получения серийного номера для демо-версии). Установка *ABBYY FineReader for Linux* сводится к запуску от имени администратора двоичного файла (*.abbyocr.bin*) из архива, скачанного с сайта проекта, и введению во время установки серийного номера.

Команда для запуска распознавания выглядит примерно так:
`abbyocr -rl Russian English -if test.png; -f RTF -of test.rtf`

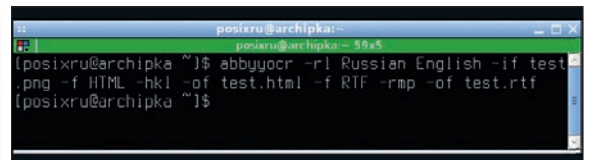
ABBYY FineReader for Linux поддерживает распознавание текста (полное и частичное) на более чем двухстах языках и предусматривает множество опций командной строки для гибкой настройки параметров распознавания.

Для сохранения результатов распознавания используются проприетарные форматы хранения текстовых документов; основной формат сохранения – RTF. Лицензия программы проприетарная, а последняя версия – 9.0. **LXF**

Что предпочесть?

Из всех представленных в данном обзоре программ для оптического распознавания тестов наиболее подходящей и удобной для русских пользователей является *Cuneiform-Qt*, благодаря поддержке русского

языка и простому и интуитивно понятному интерфейсу; однако тем, кому нужно заниматься распознаванием текстов профессионально, стоит обратить внимание на *ABBYY FineReader for Linux*.



➤ **FineReader для Linux до сих пор не имеет GUI, хотя стоит как приличный офисный пакет...**



VII Всероссийский форум Руководителей образовательных учреждений



28-29 ноября 2011 / Москва, ГК «Рэдиссон Славянская»

28-29 ноября 2011 года в отеле «Рэдиссон Славянская» (Москва) пройдет VII Всероссийский форум Руководителей образовательных учреждений.

Главная цель мероприятия – создать условия для обмена информацией и опытом среди специалистов сферы образования из всех регионов России.

В рамках Форума предполагается организовать четыре параллельных конференции, каждая из которых будет ориентирована на представителей соответствующей ступени образования:

- Конференция Руководителей учреждений дошкольного образования;
- Конференция Руководителей учреждений общего и дополнительного образования детей;
- Конференция Руководителей учреждений начального и среднего профессионального образования;
- Конференция Руководителей учреждений высшего профессионального образования.

Программа Форума будет включать Международную специализированную выставку «Индустрия образования – 2011», на которой ведущие компании и учебные заведения страны представят свои новейшие разработки в сфере информационно-компьютерных технологий, учебно-лабораторного оборудования, методических материалов и др.

Узнайте больше, посетив сайт Форума!

WWW.FORUM-EDU.RU

Организатор Форума: ГК «Открытый Диалог». Тел.: (499) 287-88-77;
e-mail: orgcom@od-group.ru; www.od-group.ru



Международная специализированная выставка
ИНДУСТРИЯ ОБРАЗОВАНИЯ – 2011

Организатор Выставки



ПРИГЛАСИТЕЛЬНЫЙ БИЛЕТ

Разделы Выставки

- Современное оборудование и средства обучения;
- Информационные технологии;
- Учебная литература и издательства;
- Мебель;
- Дистанционное обучение;
- Достижения регионов по комплексной модернизации образования.

Добро пожаловать на Выставку!

Пригласительный билет дает право на бесплатное посещение выставочной экспозиции «Индустрия образования – 2011». Выставка работает 28-29 ноября 2011 года и пройдет в залах отеля «Рэдиссон Славянская». Адрес: Москва, Площадь Европы, 2 (ближайшая станция метро – «Киевская»).

Более подробную информацию о Выставке и условиях ее посещения можно узнать по телефону (495) 287-88-77 или на сайте www.forum-edu.ru

Место проведения Выставки



Схема проезда



Для посещения Выставки вырежьте пригласительный билет и предъявите его на входе

Вход свободный / Не для продажи

YAGF



Оболочка для систем OCR

Тимур Мубаракшин рассказывает, как сделать удобнее работу по распознаванию текстов, эксплуатируя сразу два движка.

Cuneiform – в переводе с английского «клинопись»; это система оптического распознавания текстов, разработанная российской компанией Cognitive Technologies. Первоначально она разрабатывалась как коммерческий проект, поставляясь с некоторыми моделями сканеров, но после нескольких лет перерыва в разработке Cognitive Technologies проект освободила, прекратив также разработку и продажу *Cuneiform*. Данный обзор посвящен одной из графических оболочек, доступных для *Cuneiform – YAGF*. Программа распространяется под лицензией GPLv3; написал ее наш соотечественник [и постоянный автор *LXF*, – прим. ред.] Андрей Боровский.

YAGF – Yet Another Graphic Front-end for Cuneiform – предоставляет графический интерфейс для консольных программ распознавания текстов *Cuneiform* и *Tesseract* на платформе Linux. Кроме того, *YAGF* позволяет управлять сканированием изображений, импортом страниц из документов PDF, их предварительной обработкой и собственно распознаванием. *YAGF* упрощает последовательное распознавание большого числа сканированных страниц.

YAGF необходимы библиотеки *Qt 4.x* и пакет проверки орфографии *aspell*. Если вы хотите управлять сканированием изображений напрямую из *YAGF*, установите программу *XScan*. Для распознавания текста потребуется либо *cuneiform*, либо *tesseract*. Для импорта страниц из файлов PDF необходимо установить утилиту *pdftoppm* или пакет *GhostScript* (который содержит программу *gs*). *YAGF* можно установить во всех основных популярных дистрибутивах, добавив специальный репозиторий в менеджер пакетов. Описывать детальный процесс установки не имеет смысла, так как в каждом дистрибутиве свои правила установки приложений.

Работа в *YAGF* состоит из нескольких этапов:

- » Получение изображения (серии изображений) или страниц
- » Подготовка к распознаванию (если необходимо)

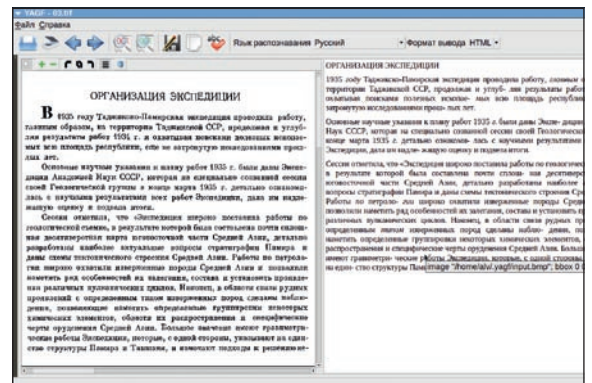
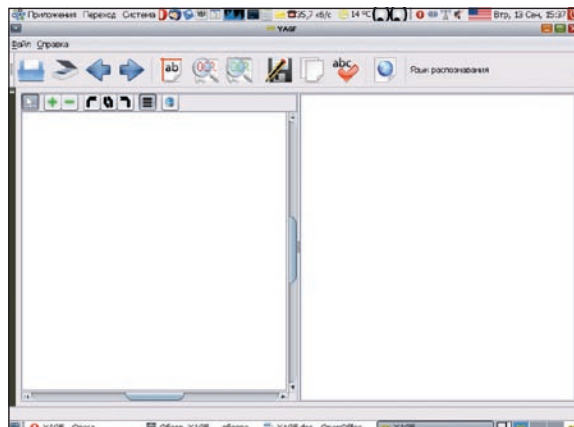
» Распознавание

» Сохранение результатов

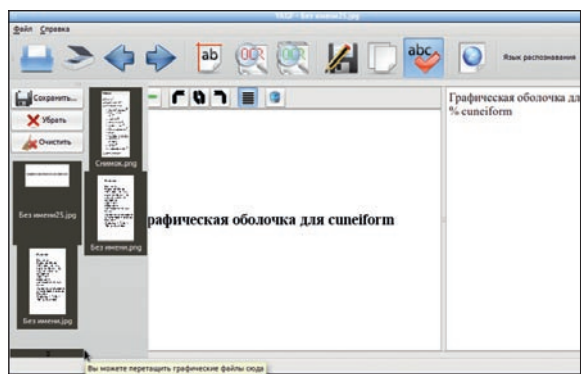
Использовать файлы изображений, сохраненные на жестком диске, или отсканировать новое изображение не составляет труда. Чтобы загрузить изображение, воспользуйтесь командой **Файл > Открыть** (можно открыть несколько файлов сразу). Вы также можете перетащить графические файлы мышью на темную полосу в левой части главного окна программы, в результате чего они будут загружены в окно приложения. *YAGF* поддерживает все основные растровые графические форматы (JPEG, PNG, BMP, TIFF, GIF, PNM, PPM, PBM и другие). Если имя открытого файла имеет вид **nameXXX.ext**, где **XXX** – последовательность цифр, вы можете переходить к предыдущему/следующему файлу с помощью кнопок перехода, расположенных на панели быстрого доступа. Например, если вы открыли файл **MyPage06.jpg**, то при щелчке кнопки перехода к следующему изображению программа попытается открыть файл **MyPage07.jpg**.

Получать изображения напрямую со сканера можно с помощью программы *XScan*. Находясь в *YAGF*, выберите **Файл > Сканировать**. Будет запущена программа *XScan*. Настройте параметры сканирования в *XScan* и нажмите кнопку «Сканировать». По окончании сканирования в окне просмотра изображений *YAGF* появится отсканированное изображение. Если вам нужно отсканировать несколько изображений, выполняйте эти операции несколько раз (в окне просмотра изображений всегда будет открыто последнее отсканированное изображение; вы можете перейти к предыдущим изображениям, используя команды перемещения). В *YAGF* можно работать, не закрывая окно *XScan*. При выходе из *YAGF* открытое программой окно *XScan* закроется автоматически. Для перехода к другим отсканированным изображениям пользуйтесь кнопками с панели быстрого доступа.

» Главное окно программы не перегружено лишними элементами.



» Работа с кириллицей в *YAGF* на высоте.



Функция пакетного распознавания документов.

Все полученные изображения отображаются в уменьшенном виде на панели изображений в левой части главного окна программы. Вы можете сохранить эти изображения в отдельную директорию с помощью кнопки «Сохранить».

Можно импортировать страницы из незашифрованных файлов PDF. Для этого воспользуйтесь командой «Файл > Импортировать из документа PDF...». В открывшемся диалоговом окне вам будет предложено выбрать файл PDF и режим экспорта (интервал страниц или весь документ). Затем вам предложат выбрать пустую директорию для сохранения импортированных страниц или создать новую.

В YAGF можно выполнять простые операции подготовки отсканированного изображения: выделение блока текста для распознавания и поворот. Если изображение ориентировано неправильно, его можно повернуть на 90 градусов по и против часовой стрелки, или на 180 градусов. Делается это с помощью кнопок панели быстрого доступа в окне просмотра изображений. Если нужно передать на распознавание не все отсканированное изображение, а его часть, необходимо выделить мышью один или несколько прямоугольных блоков в окне просмотра изображений. Если щелкнуть левой кнопкой мыши по существующему блоку, его цвет изменится на розовый. Теперь размер выбранного блока можно изменить, «ухватившись» мышью за край блока. А если щелкнуть в окне изображения правой кнопкой мыши, то появится контекстное меню, с помощью которого можно удалить все выделенные блоки, удалить выбранный блок или распознать текст выбранного блока.

Для удобства выделения блоков можно уменьшить или увеличить размеры изображения в окне просмотра (эта операция не влияет на размеры изображения, передаваемого программе *cuneiform*). Изменение видимых размеров изображения можно выполнить также с помощью комбинаций клавиш Ctrl+ и Ctrl-, или вращая колесико мыши, удерживая при этом клавишу Ctrl (точно так же можно изменить размеры шрифта в окне просмотра текста).

Если в программе открыто несколько страниц и для каждой выбраны свои ориентация и масштаб и выделены свои блоки, YAGF запомнит эти параметры для каждой страницы.

Если страница отсканирована неровно, с наклоном, можно попробовать исправить это с помощью функции «Исправить наклон страницы». Для этого предусмотрена специальная кнопка. Поскольку исходное изображение всегда сохраняется, можно вернуться к нему в случае ошибки.

Начиная с версии 0.8.7, имеется возможность выбирать программу распознавания – *cuneiform* или *tesseract* (меню «Настрой-

ки > Настройки программ распознавания»). Программа *tesseract*, начиная с версии 3.0, позволяет распознавать некоторые языки, не поддерживаемые *cuneiform* (греческий, финский и т. д.). Соответствующие языки помечены в списке языков распознавания меткой «только tesseract». Если вы выбрали *tesseract* для распознавания текстов, убедитесь, что в вашей системе установлены файлы данных для тех языков, на которых вы собираетесь выполнять распознавание.

Приступая к работе, необходимо выбрать подходящий язык распознавания (или пару языков, если распознаваемый документ написан на нескольких языках). *Cuneiform* позволяет распознавать тексты почти на всех европейских языках, а также тексты, содержащие пару языков русский–английский.

Каждый новый распознанный фрагмент текста (выделенный блок или новая страница) добавляется в редактор распознанного текста в виде нового абзаца.

По умолчанию YAGF выполняет проверку орфографии распознанного текста с помощью *libaspell*. Обычно в системе устанавливаются орфографические словари для «родной» локали системы и английского. Если вы хотите проверять орфографию для текстов на других языках, установите соответствующие словари (они должны быть в репозитории вашей системы). Если YAGF не находит нужного словаря для проверки орфографии для заданного языка распознавания, программа предупреждает вас об этом. Отключите проверку орфографии, если не хотите получать такие предупреждения.

Если нужно распознать текст сразу с нескольких изображений, можно воспользоваться пакетным распознаванием. Для этого все изображения, которые требуется распознать, должны быть открыты на панели изображений (в левой части окна программы). Щелкните кнопку «Распознать все страницы». Все от-

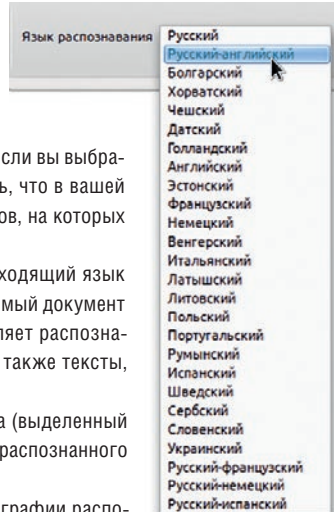
крытые изображения будут загружаться и распознаваться автоматически. При этом будет выведено диалоговое окно, отображающее ход распознавания. Вы можете остановить процесс пакетного распозна-

вания, щелкнув кнопку «Прервать». Если на распознаваемых страницах выделены блоки, будет распознан только текст внутри блоков.

Распознанный текст может быть сохранен на диске в текстовом формате (кодировка UTF-8) и в формате HTML, или скопирован в буфер обмена. Кнопка «Копировать текст в буфер обмена» заносит в буфер выделенный фрагмент распознанного текста или весь текст, если в редакторе отсутствует выделение.

Новейшая версия YAGF – 0.8.7, обновлена 29 августа 2011 года (главные новшества – поддержка *tesseract* и импорт страниц из файлов PDF).

В целом, YAGF работает быстро и с минимальным количеством ошибок, и на сегодняшний день является, пожалуй, лучшей оболочкой для OCR *Cuneiform* под Linux. **LXF**



YAGF поддерживает работу со многими европейскими языками, в том числе парные комбинации — например, «русский–английский».

Альтернативы

Как мы уже выяснили, YAGF не является единственной графической оболочкой для работы с *Cuneiform*.

Интерфейс *Cuneiform-Qt* еще более прост, но некоторые функции, предоставляемые в YAGF, в *Cuneiform-Qt* на данный момент не реализованы: это, например, касается пакетного распознавания документов.

Однако *Cuneiform-Qt* достаточно быстро работает и активно развивается. И, заметим, обе программы неплохо (по сравнению с зарубежными альтернативами) обрабатывают кириллицу, что немаловажно для русскоязычных пользователей. Интересно будет посмотреть, какой из них удастся склонить на свою сторону больше линуксоидов...

ChorusOS:

«Альтернативная история» UNIX

Роман Ярыженко перелистывает малоизвестные страницы истории *nix-систем.



Наш эксперт

Роман Ярыженко

Всегда готов потратить время и силы на изучение новой операционной системы – особенно если она распространяется вместе с исходными текстами.

Если попросить назвать микроядерную операционную систему реального времени для встраиваемых систем, известную годов с 80-х, читатель, скорее всего, ответит «QNX» – и будет прав. Но... лишь отчасти. Помимо QNX, была и, думается, и по сей день где-то применяется еще одна микроядерная ОС реального времени с открытым исходным кодом (публикуется под четырьмя открытыми лицензиями – GPL, SPL Lite, MPL и FreeBSD), ныне незаслуженно забытая. Некоторые даже считают, что QNX является подражанием данной ОС. Имя ей – ChorusOS.

История

О ранних версиях ChorusOS – тогда еще просто Chorus – удалось найти не очень много информации. Начало разработки проекта датировано 1979 годом. Тогда он был исследовательским проектом INRIA – французского Института исследований в информатике и автоматике, из стен которого, в частности, вышел Scilab – свободный аналог Matlab.

Chorus-V0, прототип ОС, разрабатывался в 1980–1982 гг. на основе трех идей. Некоторые из них мы подробнее рассмотрим позднее, пока же просто перечислим:

- » Актеры [Actors]
- » Распределенные приложения
- » Маленькое ядро [Nucleus]

Эти идеи использовались и в более поздних версиях.

V0 была написана на интерпретирующем Паскале и запущена на 8086-м процессоре.

Chorus-V1 (1982–1984) уже не была прототипом и запускалась на кластере из компьютеров SM90, базирующихся на процессорах M68K. Компьютеры были соединены 10-мегабитной Ethernet-сетью. Код на Паскале компилировался.

Ядро V2 (1984–1986) по сравнению с V1 не очень изменилось – зато внешнее API, походившее на UNIX того времени, было значительно изменено: появились функции для работы с файлами и процессами. Тем не менее, это не был UNIX – разработчики Chorus стремились к модульности (которая в те времена в UNIX отсутствовала) и к возможности распределения ресурсов между компьютерами. Эта версия была, скорее, проходной.

А вот в V3, которую начали разрабатывать в 1987 г. на коммерческой основе в компании Chorus Systems, были реализованы и новые идеи. Так, например, данная версия была переписана на C++. Разумеется, C++ был совсем не тот, который мы знаем сейчас, но по тем временам ОС, написанные на объектно-ориентированном языке, были в новинку (да, к слову, таких и сейчас мало). Также система стала именоваться микроядерной. Странная это была микроядерность... но об этом чуть позже. Также была введена возможность динамической виртуальной памяти и подкачки [demand paging], расширены некоторые базовые концепции защиты.

На базе ядра Chorus-V3 было несколько вариантов поставки.

» Chorus/Micro – голое ядро (занимало в памяти около 10 КБ) жесткого реального времени. Стоимость лицензии на разработку – 5000 долларов. На основе этого ядра даже была разработана ОС для суперкомпьютера Cray.

» Chorus/ClassiX – уже полноценный «конструктор» ОС со средой разработки.

Здесь необходимо дать некоторые пояснения. ОС и приложения для встраиваемых систем, естественно, никак не могут разрабатываться на результирующих [target] системах. Соответственно, для разработки необходима рабочая станция с соответствующим набором приложений. Но ведь на ней тоже должна стоять операционная система! В терминологии разработчиков встраиваемых систем она называется «host». Так вот, Chorus/ClassiX в качестве Host-систем поддерживал UnixWare, SunOS и Solaris. Стоимость лицензии на разработку варьировалась от 9000 до 25000 (по другим источникам, 11000) долларов, а стоимость лицензии на одно target-устройство составляла, в зависимости от функциональности, от 25 до 163 долларов для тысячи и более устройств.

» Chorus/MiX – подсистема Unix System V на базе микроядра Chorus. Требовала лицензии USL/Novell. Системные вызовы System V транслировались в системные вызовы микроядра.

Ничего не напоминает? Да-да, Wine работает на подобном же принципе! Фактически, ранние версии Windows NT были основаны на похожей идее – Native API и подсистемы. К слову, в свое время эта реализация System V считалась одной из лучших в Европе.

» Chorus/JaZZ – расширения реального времени Java на базе Chorus/ClassiX.

» Chorus/COOL – вариант с поддержкой CORBA – стандарт написания распределенных объектно-ориентированных систем повышенной сложности. По сути, это надстройка над Chorus/ClassiX.

В сентябре 1997 г. компания Chorus Systems была куплена Sun Microsystems. Возможно, это будет и не связанные события, но данная ОС через пять лет после ее покупки умрет. В пользу этой версии говорит и то, что при покупке от туда ушли ее старые разработчики (но Мишель Жьен [Michel Gien] и Марк Гийемон [Marc Guillemont] остались. После угасания ОС они даже попытаются ее реанимировать...). Но до угасания еще было далеко, и Sun успела выпустить еще несколько версий.

Sun по неизвестным причинам отказалась от поддержки Chorus/MiX. Вместо этого она сосредоточилась на развитии слоя



» Тот самый Cray, на котором стоял Chorus.

Java. Помимо этого, компания начала поставлять несколько переработанную среду разработки Chorus/ClassiX под своим брендом – Sun Embedded Workshop.

В июне 1999 г. была выпущена ChorusOS 4. Было решено отказаться от прежнего наименования систем. К сожалению, у нас нет сведений о нововведениях четвертой версии. Но имеется информация о версии 4.0.1. В числе улучшений данной версии значатся:

- » Обновлен стек TCP/IP, портированный из FreeBSD 2.2.8.
- » Упрощено администрирование.
- » Улучшены функции high-availability.
- » Введена поддержка исключений C++.

А зимой 2001 г. Sun анонсировала ChorusOS 5 и следом же, летом 2002, открыв исходные коды, отказалась от поддержки ОС. Среди новых возможностей пятой версии отметим IPv6, NTP и POSIX-RT API – расширения POSIX для поддержки приложений реального времени.

В августе (нет-нет, не 44-го! 2002-го) бывшие работники Chorus Systems, Мишель Жьен и Марк Гийемон, основали компанию Jaluna с целью вдохнуть вторую жизнь в Chorus. По неизвестной причине – не исключено, что из-за плохого маркетинга – это оказалась гальванизация трупа. Где-то в 2005 г. компанию Jaluna купила VirtualLogix, которую, в свою очередь – о, неумолимые законы капитализма! – купила RedBend Software. Быть может, ChorusOS, под иным именем, используется где-то в глубинах этой акулы капитализма, а может, и нет – сие нам неизвестно [возможно, на ее основе построен гипервизор для мобильных устройств VLX, – прим. ред.]. Если используется, то и пусть. А мы перейдем к описанию архитектуры этой достойной ОС.

Актеры, порты, LAP'ы...

Не будем рассматривать сборку из исходников ОС – это нетривиальное занятие хотя бы потому, что компиляторы сейчас, видимо, будут ругаться на те части кода, которые более ранние их (компиляторов) версии считали правильными. Да и вряд ли читателям это понадобится. Те же из них, кому вдруг захочется это сделать, вполне способны сами во всем разобраться. Итак – архитектура.

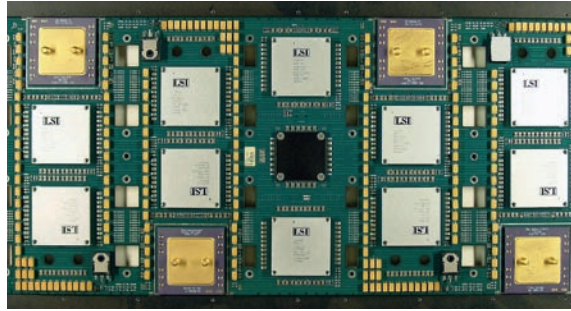
В основе ChorusOS лежит микроядро, которое сами разработчики в ранних версиях называли Nucleus. Оно состоит из модулей *kern* (реализует API микроядра), *pmm* (Persistent Memory Manager; используется при функции Hot Restart, которую мы здесь рассматривать не будем – скажем лишь, что эта функциональность задается для уменьшения времени простоя в случае сбоя) и *pd*, необходимый для обмена данными между подсистемами.

Также в составе микроядра работает Core Executive, исполнительная система, которая обеспечивает... да много чего обеспечивает, в числе прочего – обработку исключений, управление потоками, управление актерами...

Что такое «актер» [actor]? Актер – «единица инкапсуляции ресурсов» (цитата из документации). Проще говоря, это другое название процесса UNIX, с одним отличием: актер может выполняться как в адресном пространстве пользователя (при этом, естественно, адресное пространство у каждого актера свое), так и в адресном пространстве ядра [supervisor adress space] – соответственно, адресное пространство общее. Непонятно в таком случае, почему ОС – микроядерная: драйвера тоже выполняются в режиме супервизора, судя по исходным кодам; но не нам спорить с Таненбаумом. Не исключено, что при написании «Сравнения» (“A Comparison of Three Microkernels”) работа с драйверами в Chorus была реализована иначе, но это кажется сомнительным. И во избежание путаницы будем считать ее микроядерной ОС.

Потоки... тут, думаю, все понятно. Для отправки сообщений другому потоку или группе потоков используются, соответственно, порты или группа портов.

LAP (Local Access Point) и LAPSAFE применяются для вызова функций, экспортируемых актерами режима ядра [supervisor



» Процессорная плата Cray.

actors]. Различие их в том, что LAPSAFE не поддерживает экспорт напрямую: ее возможности используются для функциональности Hot Restart. Поверх микроядра лежит иерархия драйверов с двумя фреймворками – DDI (Device driver interface; нужен для сообщения между слоями драйверов) и DKI (Driver/Kernel Interface; предоставляет сервисы микроядра, специфичные для драйверов). И фреймворки, и драйвера исполняются в режиме ядра.

Далее лежит слой C_OS с актерами C_INIT (инициализация), IOM (ввод/вывод) и ADMIN (используется для специфических функций администрирования).

Актер IOM используется для реализации совместимости с POSIX. Практически же, эта подсистема основана – на чем бы вы думали? – на FreeBSD 4! Да-да, вы поняли правильно – IOM, по сути дела, является ядром FreeBSD, запущенным как актер ChorusOS, опять же, в режиме ядра (возможно, частично – мы, к сожалению, не гуру по исходным кодам данной ОС).

Наконец, слой приложений – тут, думаем, все ясно. Доступны такие системные утилиты, как *mount*, *chmod*, *route*... и даже *X11*! Из-за ограниченной поддержки эффективного и реального UID не реализован бит SUID. Но – нужна ли подобная функциональность во встраиваемых системах?..

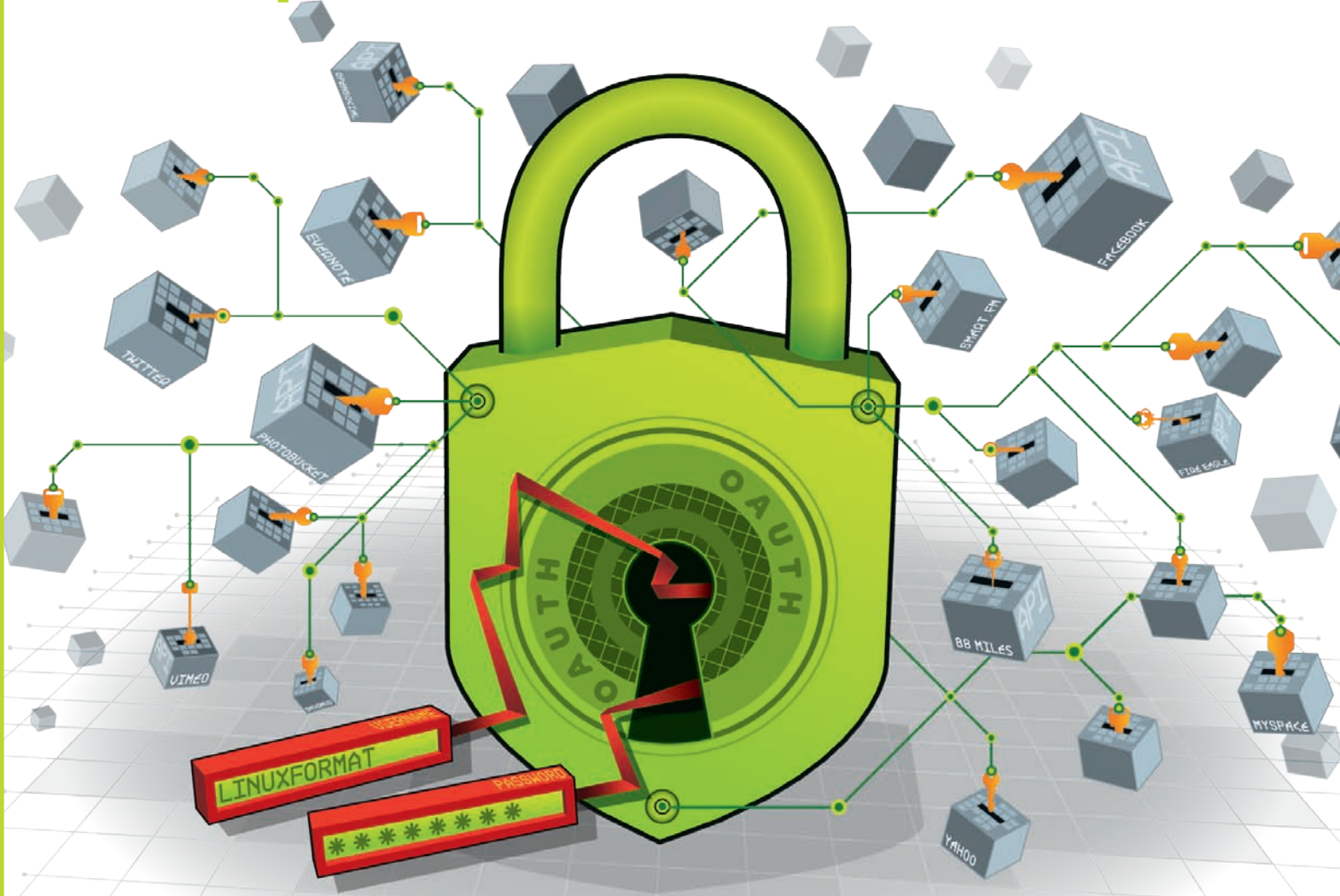
Увы, статья имеет ограниченный объем, поэтому мы вынуждены закругляться. Не были рассмотрены такие возможности, как Hot Restart, динамические библиотеки... но это неудивительно – ибо даже в книге по программированию в ChorusOS есть не все.

Итоги

ChorusOS была системой, оперед... стоп. Это уже где-то было. И все-таки – много ли микроядерных систем вышло за рамки лабораторий и имело хоть какой-нибудь коммерческий успех, пусть даже в специфической области встраиваемого оборудования? Так или иначе, ChorusOS не добилась большой известности, оставаясь в тени своей кузины QNX, и, возможно, потому и увяла. Но вдруг какой-нибудь студент университета напишет в список рассылки ChorusOS сообщение вида «Hello everybody out there using chorusos – I'm doing...»? Как знать... LXF

Ссылки по теме

- » <http://sourceforge.net/projects/jaluna/> – исходные коды, переработанные Jaluna.
- » <http://download.oracle.com/docs/cd/E19048-01/index.html> – официальная документация по ChorusOS.
- » <http://www.pps.jussieu.fr/~rifflet/PUBLICATIONS/book.ps> – «Programming under ChorusOS» (книга описывает ChorusOS 4).
- » <http://www.cs.vu.nl/pub/papers/amoeba/super95.ps.Z> – публикация Таненбаума.
- » <http://web.archive.bibalex.org/web/19970130221616/www.chorus.com/> – сайт Chorus Systems, сохраненный в архиве.
- » <http://web.archive.org/web/20010302182802/http://www.sun.com/chorusos/ds-chorusos.401.html> – Datasheet версии 4.0.1.
- » <http://web.archive.org/web/20000413075621/http://www.sun.com/chorusos/ds-chorusos3.2.html> – Datasheet версии 3.2.
- » <http://web.archive.org/web/20000815094456/http://www.sun.com/software/chorusos/wp-emb.telecom.platform/index.html> – whitepaper.
- » <http://www.leonard.nom.fr/publications/chorus/> – подборка документации по ChorusOS.



Что за штука... OAuth?

Марко Фиоретти объясняет, как сэкономить время и остаться в безопасности, авторизовав сайты поработать за вас.

В Рассказывайте, но желательно — покороче.

О OAuth (<http://OAuth.net/>/<http://OAuth.net/>) — это протокол аутентификации и авторизации, изначально разработанный для веб-приложений и возникший в недрах Twitter в 2006 году.

В Ладно, значит, это не фантомный продукт; и что он делает?

О Он позволяет стороннему ПО выступать от вашего имени, на ограниченный срок и без передачи этому ПО постоянного или полного доступа к приватной информации.

В Типа ключей, передаваемых работникам автосервиса?

О Да. С их помощью можно открыть, завести машину и проехать, но недалеко и не открывая багажник. Так и OAuth — своеобразный валет-ключ для ваших данных. Он предоставляет временный и ограниченный доступ к чему-то вашему, не отдавая при этом полного контроля.

В Так, с этим понятно, но... такая ли уж это насущная задача?

О Стала таковой, с тех пор как всевозможные онлайн-сервисы и социальные сети,

от Twitter и Flickr до интернет-банков, сделались не просто вездесущими, но и взаимосвязанными — и гораздо более полезными, когда эта взаимосвязь работает на вас.

В К примеру, если вам нужно разместить свою галерею Flickr на Facebook.

О Точно. И то, что при этом не нужно вводить все заново вручную — просто здорово. Между тем, делать это без помощи OAuth или чего-то подобного — значит предоставлять этим сайтам полный доступ ко всему вашему контенту (то есть файлам, адресной книге, или же пользованию услугами).

В Поэтому вы и упоминали и аутентификацию, и авторизацию?

О Верно. Аутентификация – это способ доказать, что вы – это действительно Вы. Заметьте, что в принципе безразлично, являетесь ли «вы» человеком или программой. Авторизация – самостоятельная и не менее важная функция. Если некто или нечто подтвердили в Facebook свою личность, это еще не повод менять мой статус от моего имени.

В А нельзя использовать для этой цели OpenID?

О OpenID – это чистая аутентификация. А OAuth, напротив, нужен в тех случаях, когда (пользуясь внутренней терминологией) некая программа (клиент) хочет получить доступ к данным от имени лица с правом авторизации (владельца ресурса), будучи совершенно не известной и независимой по отношению к программе или сервису, где хранится ресурс.

В Секунду! Это же было возможно задолго до OAuth!

О Да, но в большинстве случаев речь шла либо об использовании всего одной учетной записи в сети уже взаимодействующих сайтов, либо о предоставлении по крайней мере одному из них ваших логинов и паролей для других. OAuth пытается закрыть эту лазейку в безопасности.

В Вы говорите об авторизованном доступе к содержимому учетной записи, без сообщения моего пароля и логина?

О Предположим, вы оставили комментарий в каком-то блоге и хотите, чтобы блог переслал эту запись в Twitter от вашего имени, вместо того, чтобы перепечатывать. Когда вы сообщаете об этом блогу (например, нажав кнопку), он направляет запрос в Twitter, включающий идентификационный ключ и те данные или функции, которыми он хочет воспользоваться от вашего имени. Twitter (а не блог!) предоставляет вам пользовательскую web-форму авторизации, размещенную на его сервере. Если вы успешно авторизуетесь в Twitter и подтвердите запрос, Twitter удовлетворит запрос блога. Без передачи вашего пароля и имени пользователя.

В Да, это круто! А что будет происходить потом?

О Twitter вернет вас в браузере назад в блог, но уже со специальной ссылкой, содержащей «маркер доступа» или ключ однократной авторизации. Теперь блог сможет представить этот ключ Twitter в подтверждение того, что именно он только что получил разрешение на использование вашей учетной записи или ее изменение.

В И это будет работать на любом сайте, поддерживающем OAuth, не только на Twitter?

О Безусловно. Если, конечно, эти сайты не отклоняют первоначальный запрос. Помимо удобства для конечного пользователя, еще одним мощным стимулом для OAuth было усложнить жизнь спам-ботам и другим вредоносным программам.

В Каким образом у него может такое получиться?

О Независимо от процедуры авторизации, программа может работать как описано выше только в том случае, если это разрешено сайтом, к которому она пытается получить доступ. В OAuth это достигается путем использования нескольких идентификационных ключей, или верительных реквизитов, параллельно.

В Что это за реквизиты и кто, собственно, их выдает?

О Те, о которых мы уже говорили: данные, предоставляющие какой-либо программе доступ без выдачи пароля, называются реквизитами доступа. Но перед этим клиент дол-

«Мощным стимулом для OAuth было усложнить жизнь спам-ботам и вредоносным программам.»

жен сообщить серверу достоверные учетные данные. Как правило, они предоставляются самим web-сервером. Когда разработчики ПО желают использовать OAuth, они регистрируются на сервере для получения этих реквизитов, или ключей. Таким образом стало немного проще бороться с некоторыми вредоносными программами, но также привело к сбоям в работе многих действующих.

В Вы все время говорите о сайтах. Значит ли это, что OAuth неприменим к настольному ПО?

О Вот это каверзный вопрос. Технически, ничто не мешает клиентам OAuth быть обычными настольными приложениями внутри вашего компьютера. На практике же (по крайней мере с OAuth 1.0), это значило либо усложнить жизнь добросовестным разработчикам, либо сделать всю систему реквизитов доступа почти бесполезной. Особенно если речь идет об Open Source.

В Фу! Разумеется, хорошего тут мало, но почему это так?

О Потому что схема, которую я описал, превосходно работает в том случае, если реквизиты прописаны в исходном коде и/или скомпилированной программе, работающей только на web-сервере, где нельзя прочитать эти реквизиты в самом коде, или же, посредством шестнадцатеричных редакторов и им подобных, в исполняемых файлах программы.

В Так вот почему с открытым ПО проблема еще больше?

О Именно. Если вы включаете что-то заведомо конфиденциальное в исходный код, который может скачать и изучить любой... оно уже не конфиденциально по определению, не так ли?

В Конечно, но тут речь идет только о меньшей эффективности существующей схемы. А почему вы упоминали, что OAuth «ломает» существующее ПО?

О Потому что до OAuth 1.0, любой человек, минимально знакомый с написанием скриптов оболочки и с URL (к таковым причисляется, между прочим, ваш покорный слуга!), мог за пару минут создать нечто, позволяющее автоматически зарегистрироваться в Twitter, прочитать ленту или создать пост. С появлением OAuth это стало невозможно без предоставления достоверных, зарегистрированных реквизитов доступа. Даже если получать эти реквизиты придется дольше, чем писать скрипт!

В А нельзя ли как-то исправить эти скрипты?

О Конечно, можно: достаточно использовать всего одну из многочисленных библиотек ПО, которые уже зарегистрированы. Однако процесс написания и поддержки сценариев все равно останется более сложным, чем раньше. По крайней мере до выхода OAuth 2.0

В Ага, значит, ожидается версия 2.0? А когда?

О На данный момент, планируется, что OAuth 2.0 будет готов к концу 2011.

В Что нового в OAuth 2.0? Решит ли он эти проблемы?

О Возможно. Одним из главных изменений является добавление или переопределение так называемых «потоков» [flows] для ускорения получения реквизитов доступа, даже если клиент является не web-сервером, а, к примеру, приложением на мобильном устройстве. Предусмотрен также поток на основе cookies, позволяющий оживить старые скрипты, использующие cURL. Ожидается также и повышение производительности, поскольку OAuth 1.0 не очень хорошо масштабируется.

В Где можно получить дополнительную информацию?

О Официальное представление OAuth – на сайте <http://OAuth.net/about/> или <http://hueniverse.com/2010/05/introducing-OAuth-2-0/>.

LXF



По рецептам доктора Брауна

Д-р Крис Браун

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

**День рождения
Всемирной паутины**

В чем разница между Всемирной паутиной и Интернетом? Подозреваю, что в глазах большинства пользователей это одно и то же.

Я пишу это 20 лет спустя почти в тот же день, как Тим Бернерс-Ли [Tim Berners-Lee] опубликовал во Всемирной паутине первую страницу.

Интернету же около сорока. Задолго до появления Всемирной паутины он предоставлял инфраструктуру для маршрутизации и перемещения данных между компьютерами по всему миру: для удаленного доступа служил Telnet, для передачи файлов – FTP, для почты – SMTP и для разрешения имен – DNS.

Web всего лишь, так сказать, привнес HTML как язык разметки для написания web-страниц и HTTP как протокол для их доставки в браузер.

Все в HTTP-поезд!

Со временем HTTP начал использоваться как транспортный протокол сам по себе, наподобие TCP, и используется для переноса трафика, отличного от HTML (типа трафика SOAP/WSDL/XML, используемого web-сервисами).

Сегодня трафик HTTP на порт 80 – пожалуй, все, что можно пропихнуть через корпоративный брандмауэр или прокси.

Конечно, настоящей инновацией была гиперссылка, а бурный рост объема публично доступной информации вызвал настоящее потрясение.

Мой первый опыт общения со Всемирной паутиной состоялся на рабочей станции Sun, пожалуй, в 1993 или 1994 году.

Мы пользовались браузером NCSA Mosaic (который тогда был единственным). Уж не помню, какую именно страницу я открыл первой, но помню, как пытался запомнить синтаксис http://.

И мне не дает покоя вопрос: куда завели бы ссылки на самой первой web-странице?

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



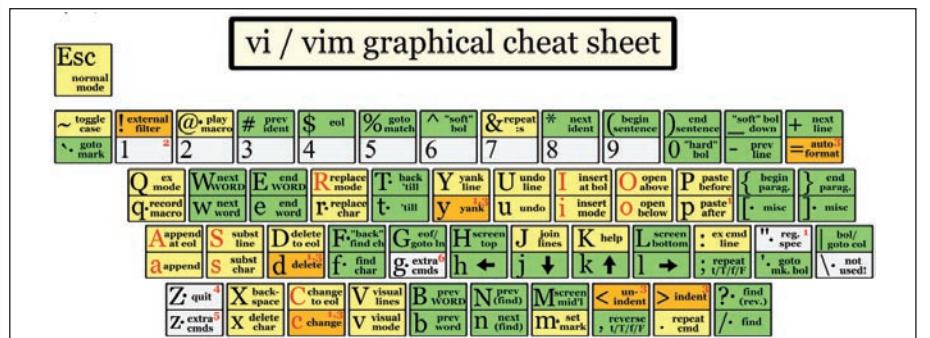
Как стать сисадмином

Восемь простых шагов к вершинам мастерства.

Недavno слушатель курса «Введение в Linux» спросил меня, как получить знания системного администратора Linux. Он имел опыт работы в Windows и изучал основы командной строки Linux. Конечно, я рекомендовал ему курс «Администратор Linux», но в его бюджете этот курс был не предусмотрен, и я сформулировал несколько советов...

- » Стиснув зубы, освоитесь в командной строке и редактировании файлов настройки вручную. Изучите *grep* и другие классические фильтры, чтобы пользоваться ими без долгих раздумий.
- » Перестаньте жаловаться на *Vi* и изучите его. Никогда не копируйте файлы настройки на флэшку, чтобы открыть их в Блокноте.
- » Читайте man-страницы. Часто они непонятны (а порой ошибочны), но все равно читайте.
- » Научитесь набирать. Бежать на курсы машинописы или набирать 60 слов в минуту не надо, но ради Бога, возвысьтесь над печатью одним пальцем.

- » Установите *VMWare* (или любой другой гипервизор), запустите несколько виртуальных машин с Linux и играйте, играйте. Создайте сайт с *Apache*, почтовый сервер с *Postfix*, DNS-сервер с *Bind*. Создайте брандмауэр на одной VM и исследуйте его с помощью *Nmap* на другой.ломайте. Находите ошибки и заставляйте работать снова. Затем повторите все это в другом дистрибутиве Linux.
- » Купите *Unix and Linux System Administration Handbook* [Справочник по системному администрированию Unix и Linux] Немет, Снайдера, Хайна и Уэйли [Nemeth, Snyder, Hein, and Whaley], 4-е издание. По мне, это лучшая книга на данную тему.
- » Помните, что поддержка безопасности системы – постоянный процесс, а не разовая акция, и уж точно не спохватывание потом.
- » Не бойтесь просить помощи у сообщества, но учитесь спрашивать правильно. Не требуйте помощи и не беситесь, если что-то с виду не работает. И всегда благодарите.



» Вы всегда подозревали, что в *Vi* каждая клавиша ЧТО-НИБУДЬ да делает. Эта удобная табличка-подсказка напомнит, что именно. Подробности см. на <http://serverfault.com/questions/31495/>.

Вопрос времени

Как в отсутствие руки и наручных часов компьютер узнает, который час? И почему это важно?

Время, сказали бы вы, дело ясное. Если держаться подальше от черных дыр и не путешествовать со скоростью света, оно идет себе со скоростью, м-м, час в час. Но на практике не все так просто.

Время в Linux, а до него – в Unix, измеряется в секундах с момента, называемого «эпохой» – первого мгновения 1970 года по Всемирному скоординированному времени (UTC).

Изначально в Unix это значение хранилось в 32-битном целом числе со знаком (тип `typedef` под названием `time_t`), и вычисления покажут вам, что оно охватывает даты с пятницы 13 декабря 1901 года до вторника 19 января 2038 года. После этого дня оно вернется обратно в 1901 год.

Оказывается, в 32-битных ядрах Linux `time_t` осталось 32-битным. Я написал об этом Линусу Торвалдсу [Linus Torvalds], и он любезно ответил: «Мы полагаем, что через 25+ лет это уже никого не взволнует. Поэтому мы даже не вникали в создание для них расширенных системных вызовов. Если через 15 лет люди все еще будут пользоваться 32-битными ядрами x86, мы можем решить, что десять-двадцать лет – уже достаточно близкий срок, чтобы заняться этим, и сделаем какое-нибудь 64-битное расширение `time_t`. Хотя я сильно сомневаюсь, что это произойдет».

В 64-битном ядре Linux `time_t` – 64-битный тип, и он способен представлять даты... ГОРАЗДО более поздние, чем те, о которых людям стоит беспокоиться.

Текущее значение времени (секунды с момента «эпохи») легко получить командой

```
$ date +%s
```

и когда я это пишу, оно равно 1310239755. Конечно, пользы от него немного – если кто-то остановит вас на улице и спросит, который час, вы вряд ли ответите: «Один миллиард триста десять миллионов двести тридцать девять тысяч семьсот пятьдесят пять».

Минуточку...

Это приводит меня к сложному вопросу о том, как записывать моменты времени в удобной для человека форме. Здесь есть традиционные сложности – високосные годы, неодинаковые длины месяцев («Тридцать дней в сентябре...»), часовые пояса, летнее время, местные культурные нормы и поправки к календарю, сделанные римским папой Григорием XIII в 1752 году. (Попробуйте набрать `cal 1752` и взглянуть на сентябрь.)

На компьютере с Linux часов на самом деле двое – аппаратные и программные. Аппаратные часы – крохотная электросхема на материнской плате. Они питаются от батарейки и должны тикать независимо от типа операционной системы и даже от включения компьютера.

Программными часами управляет ядро Linux. При загрузке системы показания программных часов первоначально задаются аппаратными, но затем аппаратные часы игнорируются. Например, при выводе времени и даты командой `date` опрашиваются программные часы. При выключении системы программные часы снова синхронизируются с аппаратными. (Подробности см. на [тап-странице hwclock](#).)

Предоставленные самим себе, аппаратные часы компьютера будут «уплывать» – пусть и на несколько секунд в день, но разница постепенно накапливается. Потеря синхронизации часов может вызывать проблемы: временные отметки файлов журналов на нескольких компьютерах рассогласовываются, запутывая про-

Пищать бессмысленно

Проблема точной синхронизации времени перед лицом задержек распространения получила интересное возрождение с приходом цифрового радиовещания.

Задержки в буферах системы цифрового радиовещания (и в самом приемнике) достигают нескольких секунд, делая нелепым секундную точность сигнала точного Гринвичского времени (шесть «писков»).

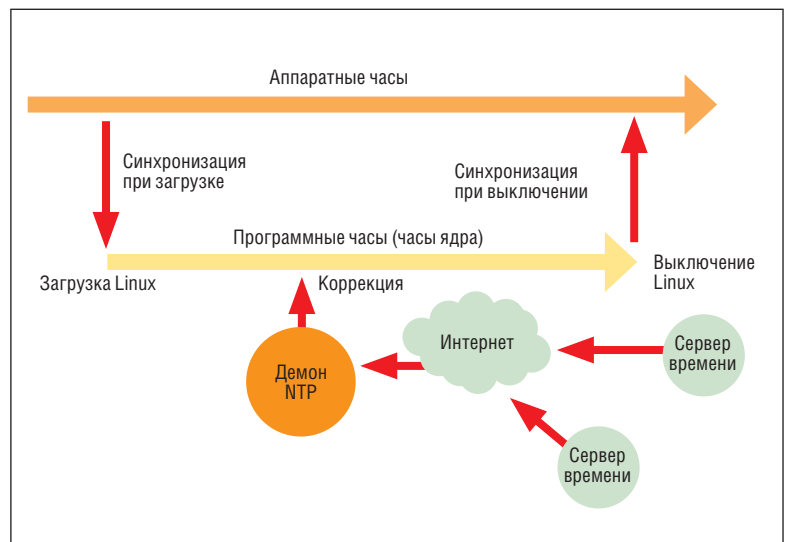
Кстати, я однажды видел в Королевской Гринвичской обсерватории создателя электрической цепи, которая формирует эти шесть «писков».

верки; временные отметки файлов на файловых серверах могут угодить в будущее по отношению к клиенту (сбивая с толку множество программ вроде *make*), а протоколы аутентификации, применяющие зашифрованные временные отметки, начнут выдавать ошибки.

Лучший метод обеспечения синхронизации часов компьютера – протокол сетевого времени (Network Time Protocol – NTP), реализуемый демоном NTP (*ntpd*). Этот умный протокол, который учитывает задержки и «дрожание», присущие пакетам, передаваемым через Интернет, и синхронизирует часы с точностью до 10 мс. Пользоваться демоном очень просто – в его файле настройки (вероятно, `/etc/ntp.conf`) надо указать несколько серверов времени, желательно не меньше трех.

Есть и более грубая утилита, под названием *ntpdate*. Она выполняет единичную коррекцию системных часов, обратившись к одному серверу времени. Если она вызывает внезапные скачки времени (особенно в обратном направлении) на вашем компьютере, это может привести к проблемам.

Цитирую ее [тап-страницу](#): «Важно учесть, что *ntpdate* с притянутыми скриптами *Cron* не заменяет демон NTP, который добивается максимальной точности и надежности благодаря сложным алгоритмам».



➤ Синхронизация аппаратных и программных часов со внешними серверами времени поддерживает точность отсчетов времени в компьютере.

Программы как сервисы

Лет через десять компьютеры будут считаться скорее устройствами доступа, чем программируемыми машинами. Присоединяйтесь к облачной революции.

Похоже, я становлюсь «проповедником облаков». На последних учебных курсах я пел хвалу облакам и обсуждал их риски с довольно скептической аудиторией. На страницах этого журнала я не раз писал об облачных вычислениях. Я показывал, как развернуть экземпляры компьютеров в облаке Amazon Web Services (**LXF127**), как пользоваться сервисом облачного сервиса вроде Rackspace (**LXF142**) и даже как создать свое личное облако с помощью Eucalyptus (**LXF148**).

Все это примеры того, что в промышленности называется «Инфраструктура как сервис» (Infrastructure as a Service – IaaS), и это лишь одна грань облачных вычислений. Сегодня я хочу поговорить о другом аспекте облачных вычислений – «Программа как сервис» (Software as a Service – SaaS).

Протицирую сайт SaaS Directory (<http://www.saasdir.co.uk/guides>): «Приложение как сервис (SaaS) – приложение, размещаемое на удаленном сервере и доступное через Интернет. Оно обычно относится к бизнес-программам и в бизнесе считается малозатратным способом получить специализированное ПО, с которым могут работать несколько пользователей. SaaS снимает с потребителя бремя обслуживания, текущего действия и поддержки ПО, тем самым снижая его затраты».

Один из самых старых и известных подобных сервисов – электронная почта в Интернете, например, Gmail. Подобные почтовые сервисы демонстрируют ключевые возможности многих SaaS-продуктов:

- » У них есть web-интерфейс (на основе браузера).
- » Доступ к ним возможен через браузер отовсюду, где есть компьютер (или планшет, или смартфон) и интернет-соединение. Установка дополнительного ПО не требуется.
- » Данные для приложения (в данном случае, ваши письма) хранятся на облаке, а не на вашем компьютере.
- » Сервисы бесплатны для мелкомасштабного личного пользования. Нет «платежного барьера», лишь преодолев который, вы получите доступ к сервису.

Однако электронная почта с web-интерфейсом – лишь верхушка огромного айсберга SaaS. Если у вас есть полчаса, зайдите в свою учетную запись Gmail (а если у вас таковой нет, поступите своим самолюбием и заведите ее) и исследуйте другие программы, к которым она дает доступ. Google Docs (где я и пишу этот текст) предоставляют Google-эквивалент текстового ре-

» Тренды Google показывают резкий рост интереса к облачным вычислениям (измеряемый в объеме поисковых запросов) за последние два года.



Чья война?

Раньше я думал, что война Open Source будет, так сказать, «выиграна», если число компьютеров с Linux превысит число компьютеров с Windows. Но поскольку мы все чаще обращаемся к облачным приложениям, тип операционной системы на локальном компьютере становится все менее важным. Теперь я думаю, что проблема исчезнет из-за своей незначительности. Людям будет просто все равно.

доктора, программы для работы с презентациями и электронных таблиц, которые мы обычно берем из *OpenOffice.org* (или *Microsoft Word/PowerPoint/Excel*). Возможности редактирования не так богаты, как в *Office* или *Word*, но особыми сверхвозможностями я и в тех не пользуюсь. Календарь Google – обычный календарь и программа для управления событиями. Затем, конечно, Picasa – программа редактирования и размещения фотографий в Web.

Эффективное SaaS-приложение от Google – Google Translate. Здесь можно набрать текст на любом из почти 60 языков, и он будет переведен на любой из остальных 59. Я не настолько силен в иностранных языках, чтобы судить о качестве результата, но выглядит сервис впечатляюще.

Большой плюс Google Docs (и SaaS в целом) в том, что они облегчают совместную работу. С любым владельцем учетной записи в Google легко открыть общий доступ к документу – с правами редактирования, и в режиме просмотра. Отпадают пересылки туда-сюда по электронной почте и проверки, что у всех правильная версия – версия теперь только одна.

Существуют буквально сотни других поставщиков приложений SaaS, предоставляющих огромный набор программ, способных обеспечить работу целой компании. Опять же зайдите на www.saasdir.co.uk и посмотрите на список. Существуют программы для ведения счетов, бухгалтерского учета, управления взаимодействием с клиентами (CRM – Customer Relation Management), управления кадрами (наем, обучение, оценка производительности) и многого другого.

Например, зайдите на zoho.com. Здесь имеется набор программ, похожих на программы Google: календари, планировщики, хранилища документов и почты и огромный набор более конкретных бизнес-приложений. Опять же, все это бесплатно для начинающих, и платить нужно только за расширение.

Что в хранилище?

«Хранилище как сервис» пока не официальный термин в облачных вычислениях, хотя его стоило бы сделать таковым. Существует множество сервисов, предлагающих место для хранения файлов в облаке.

» Хорошо известен Dropbox (<http://dropbox.com>). Он доступен для бесплатного использования (до 2 ГБ места), далее \$9,99 в месяц за 50 ГБ и т. д. Основное назначение Dropbox – синхронизация файловых систем между различными компьютерами – настольным, ноутбуком, телефоном и т. д.

» Mozy (<http://mozy.co.uk>) – другое популярное хранилище, но с акцентом на резервное копирование, а не синхронизацию. Здесь предлагается шифрование файлов пользователя. Mozy принадлежит компании EMC, поэтому должно хорошо справляться со своими задачами.

» Пользователям Ubuntu, возможно, знаком Ubuntu One (<https://one.ubuntu.com>). Он предлагает бесплатно 5 ГБ дискового пространства. Жестко нацеленный на потребительский рынок Ubuntu One позволяет хранить свою музыку и закачивать ее в смартфон.

» S3 (simple storage service – простой сервис хранилища) от Amazon – более «инфраструктурный» сервис по сравнению с сервисами, ориентированными на конечного пользователя, как Dropbox или Mozy. S3 предлагает место в «корзинах» с репликацией, «обеспечивающей 99,99999999 % непрерывность доступа и 99,99 % доступность объектов в течение года». То есть каждый год можно потерять до одного объекта из миллиарда и лишиться доступа к своим объектам не более чем на час. Бесплатно предоставляется 5 ГБ, а за деньги – до 5000 ТБ (см. aws.amazon.com/s3). Также оплачивается трафик, который бывает трудно оценить.

Немного — бесплатно

В мире открытого ПО мы привыкли получать богатый функционал бесплатно – *OpenOffice.org*, *GIMP*, *Audacity* и сотни других приложений. Конечно, это сильно отличается от модели, которая кормит традиционных производителей ПО, когда за все нужно платить вперед: нет денег – нет программы. Мир SaaS предлагает модель оплаты, которая лежит где-то посередине между свободным ПО и традиционными моделями. Мелкомасштабное использование часто бесплатно; вы платите только тогда, когда хотите увеличить количество пользователей, объем хранилища, повысить скорость выполнения транзакций и т.д.

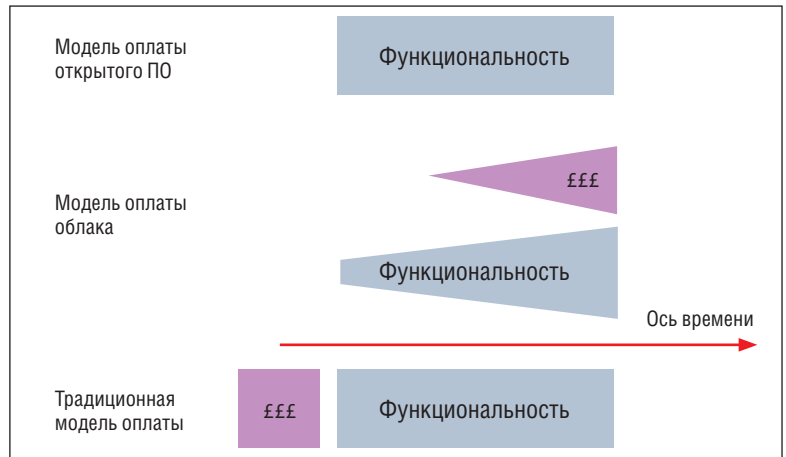
Возможностей бесплатной версии вполне хватает, чтобы понять, подходит ли она вам. Небольшим компаниям их может хватить даже для постоянной работы. Другие сервисы (например, Google Apps для бизнес-использования) предлагают бесплатные версии на время ознакомления.

Многие пользователи и компании боятся доверять свои документы и данные облачному провайдеру. Некоторые из этих страхов серьезно обоснованы. Например, существует закон, ограничивающий (географически) места, где можно хранить медицинские карты пациентов. Если вы доверите эти данные облаку, труднее будет доказать, что вы не нарушаете закон.

Других просто беспокоит смутное понятие «безопасности», под которой они обычно понимают конфиденциальность их данных. И еще мне попадались люди, которые решительно отказывались хранить данные «в облаке», боясь утечек.

Хотя эти риски нельзя полностью исключить, их нужно рассматривать в контексте. Большинство из нас хранят деньги в банке, а не в коробке под кроватью. Мы доверяем банку заботу о них и считаем, что там они в большей безопасности. Есть и другие примеры доверия – мы даем номера наших кредитных карт продавцам, когда что-то покупаем. Мы доверяем им.

Большинство облачных провайдеров очень ответственно подходят к безопасности, и во многих случаях их системы защищены



лучше, чем в дата-центрах, и явно лучше, чем пара web-серверов в офисе, обслуживаемых дилетантом. Например, web-сервисы Amazon Web Services сертифицированы PCI DSS Level 1, ISO 27001 и SAS 70 Type (сообщаю для тех, кто, в отличие от меня, знает, что это такое).

Есть и другие причины, по которым облачные вычисления на самом деле могут повысить, а не снизить безопасность. Есть новое поколение компьютеров (например, Chromebook) – чисто облачных устройств, в которых данные не хранятся локально и на которые нельзя устанавливать приложения. Такой компьютер (предположительно) не заразишь вирусами, и если его украдут, то воры не заполучат никакие важные локальные данные.

Вторая по значимости причина беспокойства после конфиденциальности – получение доступа. Существует риск краткосрочной недоступности данных (хотя в сервисных соглашениях облачных провайдеров указывается доступность в 99,95 % – это всего четыре часа простоя в год). Конечно, в сельской местности отсутствие широкополосного доступа или 3G-сигнала сводит доступность данных к нулю. Затем, есть и моя любимая мозоль – ненадежное интернет-соединение в большинстве отелей.

Власть — народу

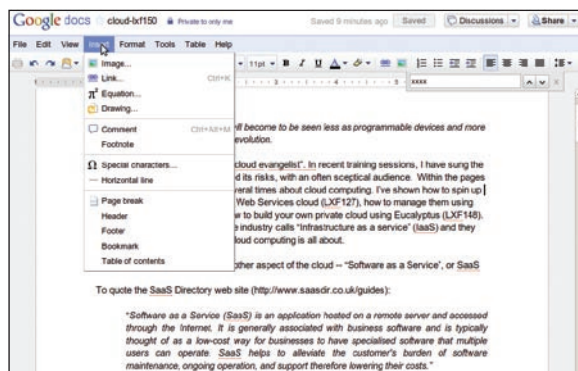
Впрочем, самое большое опасение – что облачный провайдер, на чьих серверах хранились данные вашей компании, вдруг прекратит свою деятельность. Трудно ли будет перенести данные к другому провайдеру? Сможете ли вы вообще до них добраться? В традиционной модели с покупкой программ и их локальной установкой они по крайней мере не перестанут работать, если компания-производитель обанкротится.

Приведу аналогичный пример. Когда электричеством начали снабжать заводы (и жилые дома), на каждом заводе стоял собственный генератор. Позже расширилась сеть местных электростанций. Легко представить, как сопротивлялись некоторые владельцы заводов этому переходу. Сегодня мы получаем все электричество из национальной сети, нимало не беспокоясь по этому поводу. Век спустя, когда в массы начинают приходить облачные вычисления, ситуация очень похожа на предыдущую.

Боссам, воротящим нос от облачных технологий, скорее всего придется принять их по тем же причинам, по которым предыдущие боссы в конце концов стали покупать электричество у электростанций: это дешевле.

Если вы хотите поближе познакомиться с облачными вычислениями, я от души советую вам «*Краткое руководство по облачным вычислениям*» [A Brief Guide to Cloud Computing] Кристофера Барнетта [Christopher Barnatt]. Книга написана понятным, не техническим языком, и ее можно прочесть за день. Она дает прекрасное представление об облачных вычислениях и направлении их развития. LXF

» В традиционных моделях приобретения ПО нужна предоплата. В облачных моделях вы пользуетесь программами даром, и платите только при расширении.



» Google Docs предоставляют богатую среду редактирования документов и презентаций на базе облака.

softline®



Services

Software

Cloud

ИТ-архитектура вашего бизнеса



SaaS

CLOUD

SOLUTIONS

SAM

юридическая
поддержка

remote workplace

PROJECTS

VIRTUALIZATION

LINUX

программное
обеспечение

education

hardware

legal support

Mirapolis

ERP&CRM

защита
персональных данных

САПР и ГИС

SERVICES

информационная
безопасность

SECURITY

учебный
центр

CAD
CAM
CAE

SOFTWARE

техническая поддержка

search

flexible

CONSULTING

data leak
prevention

корпоративный портал
DeskWork

open
source

PLATFORM

protection

hosting

technical support

решения Microsoft

+7 (495) 232-00-23

www.softline.ru

info@softline.ru



Наши эксперты помогут вам с любым приложением Linux!



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

Невежество атакует

А Баба-Яга против!
Баба-Яга

Жили-были честные мошенники Astrolabe. Делали и продавали мошенникам рангом ниже литературу и программы с целью обмана и надувательства. А большинство человечества и радо обманываться и думать, что далекие звезды влияют на жизнь на Земле. Кипело дело и плавилось. Никто особо не возражал: мошенники вроде безобидные, мошенничество проверено веками – ну и пусть их.

Но безобидных-то мошенников не бывает, и наши были не исключение: затеяли они уже современное мошенничество. Купили они права на старое справочное издание «The American Atlas» и подали в суд на их нарушение. А ответчиком назначили Артура Олсона [Arthur David Olson], который создал и поддерживал открытую базу часовых поясов. Он посмел прочитать справочник и поделиться содержащейся там информацией с общественностью. Но это победы. Главное – он указал источник получения общеизвестной информации (тот самый «The American Atlas»). За что и поплатился.

Будь эта база нужна только Олсону, можно было бы скрипнуть зубами и заречься впредь приближаться к гороскопам и их производным, но для большинства Unix-подобных систем (включая GNU/Linux) эта база была центральной. От действий, казалось бы, мелких мошенников неудобство широким массам.

Мораль: Мошенничество – зло и должно всячески изничтожаться, независимо от методики мошенников.

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Защищать себя 64

Осваиваем пароли, шифрование файлов и построение брандмауэров: простые способы защиты вашей системы.



Управлять

томами 68

Ходы и выходы продвинутого разбиения диска при помощи Менеджера логических томов от Fedora.



Создавать

свою Wiki 72

Быстрое и эффективное построение web-страниц с использованием мощного ПО MediaWiki.



Писать

в тетрадку 76

Профилируем многочисленные словарные значения, оставив только те, что нам нужны.



Вникать

в Erlang 80

Последний парад наступил: неосвоенных базовых особенностей языка уже не осталось. Впереди – тонкости.



Объединять

машины 84

У вас больше двух компьютеров? Свяжите их в единое целое под говорящим именем Synergy.

Совет месяца: Установка пакетов



Как правило, лучше всего устанавливать программы через менеджер пакетов вашего дистрибутива; но иногда нужной вам программы или версии в репозитории не оказывается.

Беде помогает установка из исходников, но она чревата проблемами: в вашей системе появляются файлы, не известные менеджеру пакетов, и не во всех пакетах с исходными текстами предусматривается опция деинсталляции.

Решение – *Checkinstall*, созданный для использования со стандартным методом построения из исходников *Autotools* путем `./configure, make и make install`.

Checkinstall заменяет последний шаг: вместо установки файлов в файловой системе, он строит из них пакет, а заодно может его и установить, так что файлы попадают под контроль менеджера пакетов.

Надо только, как и раньше, запустить `configure` и `make`, а затем вместо `make install` скомандовать `checkinstall`.

Если код использует для установки не `make install`, а другую команду, припишите ее следом за `checkinstall` – например, `checkinstall ./install.sh`

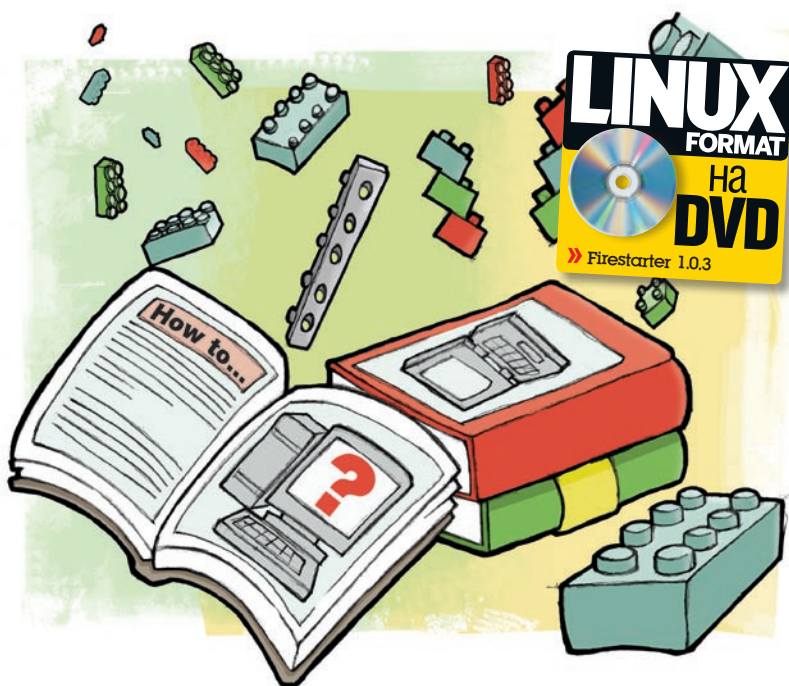
Checkinstall запросит тип пакета для создания, а также некоторые другие опции; или же можно указать их в командной строке ключом `--type`. Например, на Ubuntu понадобится выполнить следующую команду:

```
sudo checkinstall --type=debian --install=yes
```

В файле `checkinstallrc` также можно задать значения по умолчанию. Теперь вы запросто сумеете внедрить в жизнь исходные коды всего того добра, которое Майк запикивает на DVD.

Защита: Укрепим

Оградите себя от онлайн-угроз, выбрав подходящее оружие борьбы против похитителей данных – **Боб Мосс** ведет мастер-класс по безопасности.



» Смените свой старенький, истрепанный пароль на новый и гораздо более безопасный.



Наш эксперт

Боб Мосс

Разрывается между исследованием Европы и обучением по курсу компьютерных технологий.

После шумевших пробоев безопасности у Sony и Facebook высокопоставленные чиновники от промышленности обоснованно тревожатся: кто же подвергнется очередной атаке? И это касается не только правительства и корпораций – вам тоже легко пасть жертвой мошенничества и хитрости, о которых пишут в новостях, если хакер решит атаковать вашу систему. Атаке не обязательно целиться лично в вас, потому что хулиганы-скриптописцы и жулики уже засорили Интернет троянами, вирусами, шпионскими программами и фишингом. Суньтесь в петлю, и ловушка захлопнется, выдав преступникам доступ к вашим банковским счетам, паролям и конфиденциальным данным. А поскольку физически мошенники могут нахо-

диться в любой точке мира, полиция зачастую будет бессильна помочь вашей беде.

Однако, пока вы в ужасе не сбежали из Интернета, рад вам сообщить, что можно предпринять несколько простых шагов, чтобы отогнать вору – онлайн и оффлайн – желающих прорваться на вашу машину. Мы не обещаем уберечь вас от опасности прямых атак, но, надеюсь, предотвратим самые распространенные из них.

Мастер паролей

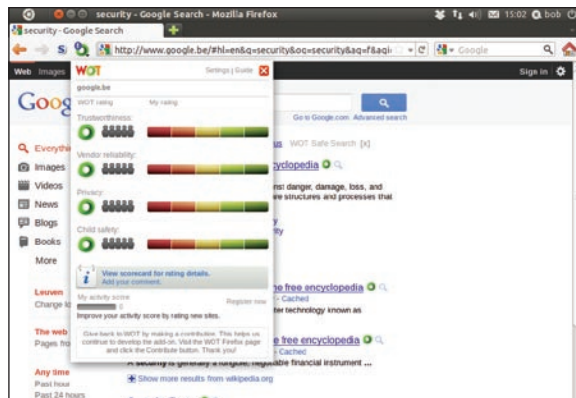
Если в вашем пароле менее шести символов и он не является смесью букв в верхнем и нижнем регистре, цифр и символов, а вы его используете для всего и никогда не меняете, это позор.

Конечно, пароли трудно запоминать. Лучшие из них, как правило, употребляют памятные слова, которые вы изменили согласно правилам – по типу "f3@Th3r" (где «e» становится 3, @ заменяет «a» и t – это заглавная Т). Однако держать по отдельному паролю для каждого web-сервиса и для входа на каждый компьютер не всегда практично.

Хорошей идеей будет использовать расширения, подобные LastPass для web-служб (<http://lastpass.com>): оно шифрует пароли и дает к ним доступ через один мастер-пароль. Потом достаточно периодически менять пароль компьютера и мастер-пароль.

Чтобы изменить пароль на дистрибутиве Linux – скажем, Ubuntu – введите Users в вашем апплете поиска приложений и нажмите Пользователи и группы. Выберите имя пользователя в левой панели и нажмите кнопку Изменить рядом с паролем. Через это окно можно даже поручить Ubuntu сгенерировать случайный пароль для вас, хотя, по понятным причинам, вы должны убедиться, что способны его запомнить.

» Почему бы не оборонить Firefox от троянов, вредоносных скриптов и фишинг-атак?



» **Месяц назад** Мы делали резервные копии данных локально и в облаке.

рабочий стол

Антивирус на Linux?

Вам могут заявить, что Linux-машины не нуждаются в антивирусном программном обеспечении, но несмотря на то, что число известных основных уязвимостей, вероятно, можно пересчитать по пальцам одной руки, нельзя сказать, что вы неуязвимы.

Всегда разумно установить антивирусную программу, такую как *ClamAV*. Для удобства пользователей Ubuntu мы добавим новые PPA, чтобы *ClamAV* был с последними обновлениями компонентов и вирусов. Перей-

дите в Центр Ubuntu Software, затем нажмите кнопку Редактировать > Источники ПО и добавьте в качестве стороннего репозитория:

```
ppa:ubuntu-clamav/ppa
```

Когда кэш обновится (вы увидите это в правой части окна), найдите пакет под названием **Virus Scanner**. Установите его и **clamav-daemon** (этот последний пакет, позволяющий *ClamAV* запускаться в фоновом режиме, вероятно, нужно установить через терминал обычным порядком).

Для запуска сканирования просто запустите GUI-приложение *Virus Scanner* и щелкните по соответствующей опции в окне. Также важно включить демон *ClamAV*, командой `sudo clamd`

Вы должны запустить обновление вирусных сигнатур и полное сканирование системы, что можно делать вручную, по крайней мере раз в день. Однако, в отличие от нашего предыдущего урока по *ClamAV* (подписчики могут найти его в **LXF131** на сайте

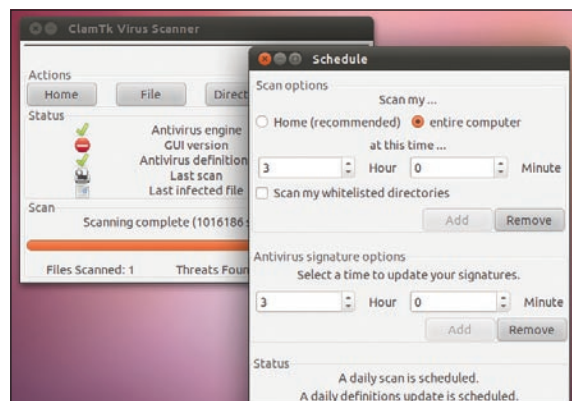
Linux Format), теперь предусмотрен запуск этих задач по расписанию, через только что установленный клиент с графическим интерфейсом. Перейдите на Дополнительно > Расписание и просто установите плановое время выполнения этих задач – например, где-то около 3 часов утра. Тогда при первом вашем входе в компьютер будут обновляться определения вирусов и сканироваться система, без малейшего вмешательства с вашей стороны.

Ключики-замочки

Но надежные пароли помогут вам только до поры. Вы, например, спрячетесь от хакеров начального уровня, отключив вещание SSID в беспроводной сети, а если им все-таки удастся засечь вашу машину и взломать пароль, все ваши файлы в их власти.

По этой причине целесообразно шифровать конфиденциальные файлы и папки (и даже целые разделы `/home`, если мошенники имеют физический доступ к вашему компьютеру).

Для этого мы применим *GnuPG* — пользователи Ubuntu найдут его клиента в меню Пароли и ключи шифрования. Здесь можно создать свой ключ шифрования PGP, а затем сделать его резервную копию на внешнем накопителе (желательно на случай утраты пароля или необходимости переформатировать диск). После настройки вы должны увидеть пункт меню Шифровать, который появляется при нажатии правой кнопки мыши на файлах, показываемых в менеджере файлов.



» Включите в расписание задачи *ClamAV*, и они будут молча работать в фоновом режиме, сокращая появляющихся аггессоров.

Скорая помощь

Знание – сила, поэтому держитесь в курсе последних угроз. Прекрасным местом для старта будут www.us-cert.gov/current, <http://on.fb.me/mR-PVOY> и <http://bit.ly/rrRw0h>.

Firefox — сокрушитель врагов

Мы часто слышим, что продукты с открытым кодом, такие как *Firefox* от Mozilla по своей природе безопасны благодаря быстрому выпуску исправлений, но это не защитит вас от всех угроз в Интернет. К счастью, мы можем бороться с этим при помощи некоторых изменений в настройках браузера и умных дополнений.

Перейдите в Редактировать > Настройки; во вкладке Содержание можно отключить автоматическую загрузку изображений (рискованно в некоторых сценариях) и JavaScript. Мы пока оставим последний включенным, поскольку вскоре перейдем

к дополнениям, ведущим борьбу с уязвимостями JavaScript. Перейдите на вкладку Безопасность; вы должны обнаружить, что все варианты отмечены, кроме мастер-пароля. Можно установить этот параметр, чтобы пользователи с физическим доступом к вашей машине не могли войти в ваши учетные записи в Интернет.

Большинство проблем с безопасностью, как правило, связаны с JavaScript или Flash-эксплоитами. Зайдя на <http://addons.mozilla.org>, вы найдете два полезных расширения – *NoScript* и *FlashBlock*. Первый находится в строке статуса вашего

браузера и по умолчанию отключает все. Нужно просто указать сайты, которым вы доверяете. Вначале это придется делать регулярно, но со временем вы обнаружите, что он действует как брандмауэр, который блокирует вредоносные скрипты, пропуская скрипты только с доверенных сайтов.

FlashBlock позволяет определить исключения с самого начала (например, YouTube).

Другое полезное расширение – *WOT* (сеть доверия – Web of Trust). Оно позволяет пользователям отметить сайты красным, желтым или зеле-

ным, в зависимости от их надежности. Оно ставит индикатор рядом со всеми ссылками в браузере и даже не позволит зайти на страницы, не внушающие доверия вообще. Единственный недостаток данного сервиса – медлительность при восстановлении сайтов, которые убрали свои вредоносные элементы.

И, наконец, расширение, о котором скажет почти каждый обычный пользователь *Firefox*: *Adblock Plus*. Оно молча убирает всякого рода рекламу с посещаемых вами сайтов, а также расширяет блокировку всплывающих окон в *Firefox*.

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

Защита: Вредные

Несколько простых процедур защитят вашу систему от атак.

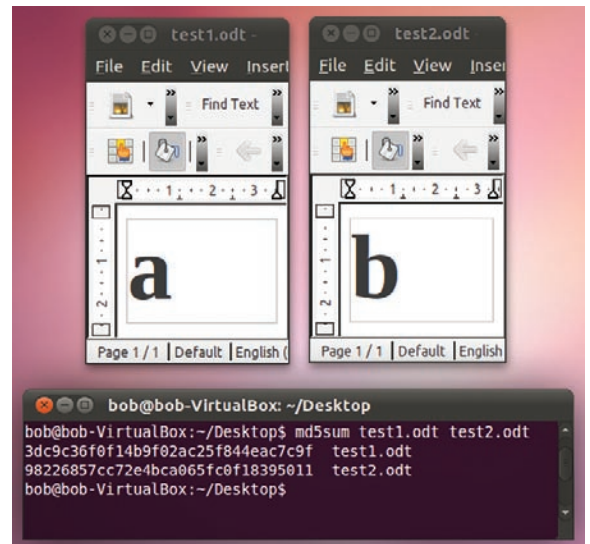
До сих пор этот учебник рассматривал компоненты, добавляемые в систему, чтобы ее обезопасить. Но из виду нередко упускается существующее и характерное поведение самого пользователя.

Независимо от созданных преград, вы все равно можете ухитриться скачать порченные файлы и папки и заразить свою систему. Мы уже рассмотрели, почему важны установка надежного пароля и проверка целостности web-страницы, а наш следующий пункт очень прост: внимательно прислушивайтесь к тому, что говорит вам ваша система.

Если ваш антивирус сообщает, что нечто требует срочного внимания, сразу этим займитесь. Затем убедитесь, что установлены все имеющиеся последние обновления системы безопасности вашего дистрибутива. Пользователи Ubuntu могут установить Центр Ubuntu Software для загрузки и установки обновлений безопасности (оставив обновления программ как дополнительное упражнение). Просто откройте программу, перейдите в Редактировать > Источники Программного обеспечения и в Обновление отметьте возможность установки обновлений безопасности без подтверждения.

Оплот целостности

Далее, обязательно проверяйте целостность файлов. С контрольными суммами MD5 вы обычно сталкиваетесь при скачивании дистрибутива Linux, так как это хороший способ гарантировать правильность. Тем не менее, они пригодятся и для проверки, что скачанный файл – тот самый, что представлен разработчика-



➤ Простые контрольные суммы помогут избежать файлов, испорченных мошенниками.

ми, и что файл или папка не были изменены злонамеренно с момента последнего доступа к нему.

Два файла (например, копия и оригинал важного документа) сравниваются командой

```
md5sum file1.odt file2.odt
```

Она проверяет документы на идентичность. Для сравнения файлов с предыдущей датой понадобится экспорт хэша MD5 (знание, созданное в файле) с помощью

```
md5sum file1.odt file2.odt > files.odt.md5
```

Для важных файлов и папок желательно делать это периодически, убеждаясь, что их никто не портил. Чтобы проверить наличие вторжения, запустите:

```
md5sum -c files.odt.md5
```

Вывод покажет 'OK' для каждого файла, включенного вами в хэш-файл (в данном случае, важный файл и его копия). Если что-либо было изменено, значение хэша станет совершенно другим, и вы будете предупреждены соответственно. Это не полная система обнаружения вторжений, но на сегодняшний день самая простая.

Последний и самый важный пункт – обеспечить регулярное резервное копирование всех важных файлов (рассматривалось в учебнике прошлого месяца, доступном подписчикам на сайте Linux Format). Мы знаем, что делать это нудно, но надо. Если ваша система скомпрометирована, то пока вы не проверите файлы по их первоначальным хэшам MD5, нельзя будет сказать, были ли они инфицированы, то есть вы по сути потеряли их. Как и защита от наводнения, пожара и кражи, резервное копирование также очень важно, если вы хотите смягчить ущерб от нарушения безопасности. **LXF**



Если вы не прочь приложить руку к настройкам вашей системы и управлять привилегиями для ваших приложений и файлов, то подписчики могут обратиться к учебнику SELinux из LXF130 с сайта Linux Format.



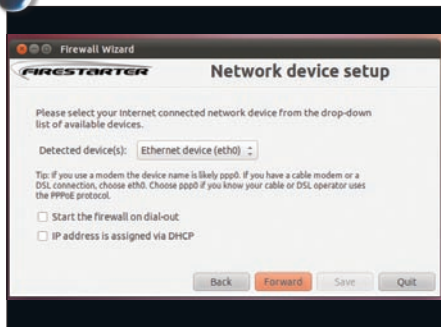
➤ Воздвигните брандмауэр, чтобы предотвратить мародерство в вашей системе.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

ПРИВЫЧКИ ДОЛОЙ



Шаг за шагом: Установим и настроим Firestarter



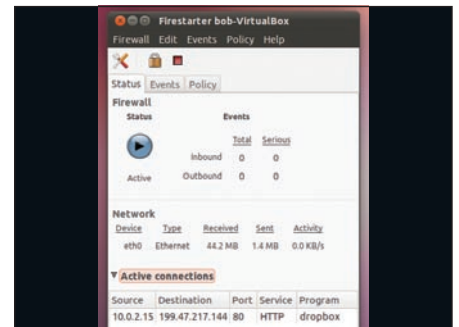
1 Запустите мастер

При первом запуске *Firestarter* выберите тип соединения и способ присвоения вам IP-адреса (DHCP, если вы используете маршрутизатор Wi-Fi).



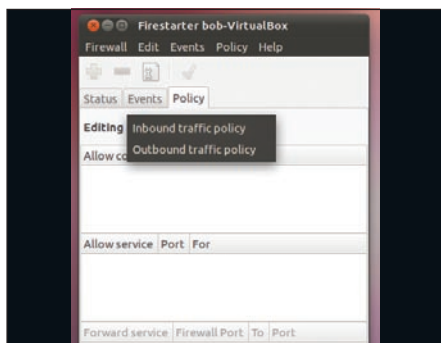
2 Поделитесь соединением

Если вы хотите, чтобы другие устройства в сети подключались к компьютеру по случаю соединения, установите флажок *Enable Internet Connection Sharing*.



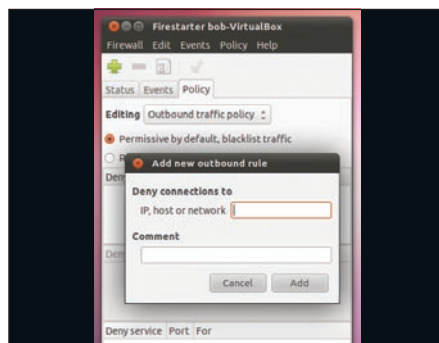
3 Окно статуса

Здесь отображается текущее состояние брандмауэра и все активные в данный момент соединения.



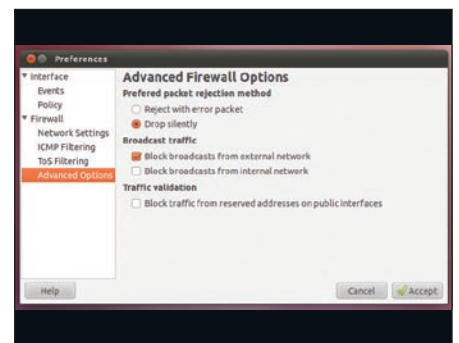
4 Задайте политики

Здесь вы можете решить, следует ли добавить или удалить правила для входящих и исходящих соединений к и из вашей машины.



5 Добавьте правило

Выбрав тот или иной вариант, можно в явном виде поместить в белый или в черный список IP-адрес или домен.



6 Изменение предпочтений

Когда вы запретите заблокированные соединения, обнаруженные во вкладке *События*, можно настроить *Firestarter* через Редактировать

Несложно о брандмауэре

Итак, что такое брандмауэр? Проще говоря, он предотвращает нежелательные соединения на и от вашей машины.

Например, входящие соединения могут содержать вредоносный вирус, или вирус мог уже сидеть в вашей системе и добрых две недели рассылаться с каждым нажатием клавиши через исходящее соединение.

Без брандмауэра вашу систему легко взломать, вызвать переполнение буфера путем создания большего количества соединений, чем ваша машина может потянуть, и вообще приме-

нить всю гамму подлых методов взлома вашей системы.

Таким образом, несмотря на веские причины наличия программного брандмауэра в вашей системе, мы здесь рассмотрим, как установить *ufw*, брандмауэр, имеющий настольные клиенты и под Gnome, и под KDE.

Пользователи Ubuntu найдут GUI-клиент под названием *Firewall Configuration* через центр Ubuntu Software; он должен удовлетворить практически все ваши потребности настройки брандмауэра. После запуска разблокируйте интерфейс

с помощью вашего пароля, чтобы делать какие-либо изменения.

Первым делом установите флажок *Включен*. По умолчанию входящие соединения запрещены, а исходящие соединения всегда разрешены. Хотя это подходит большинству пользователей, бывают случаи, когда нужно разрешить входящие соединения (например, удаленные входы для веб-страниц, использование клиентов веб-почты, VoIP и т.д.). Нажав кнопку *Добавить правило*, вы увидите целый ряд предустановленных параметров, которые можно использовать

на первой вкладке (то есть, чтобы разрешить входящие соединения для Skype и torrent-клиента *Transmission*).

Две другие вкладки позволяют блокировать конкретные адреса IP и TCP/UDP портов.

Если вы предпочитаете более тонкие настройки или хотите настроить множество других протоколов, не требуемых для обычных пользователей компьютеров, такие как логины SSH, и т.д., взгляните на наше руководство по пошаговой установке *Firestarter*: в нем учтены все эти расширенные возможности.

» Через месяц Изготовим PDF и поработаем в *LibreOffice*.

LVM: Управляем

Много лет боявшийся экрана разбиения диска в Fedora Джонатан Робертс прогнал страхи и увидел, что LVM проще, чем он думал.



Наш
эксперт

Джонатан
Робертс

Занимается Linux с юных лет, но только теперь набрался смелости взглянуть на экран разбиения диска Fedora.

При установке дистрибутивов на базе Fedora у меня годами были проблемы с экраном разбиения диска. Что, черт побери, значили «группы томов LVM», `vg_lvm` и `lv_root`? Почему Fedora не использует добрые старые ссылки на устройства и разделы, как другие дистрибутивы?

Не трата времени на познание темного смысла этих ссылок, я просто стирал настройки по умолчанию и создавал свою собственную схему. Так было до сих пор.

Но недавно, проведя несколько дней за изучением основ LVM, я оценил, насколько это мощная и полезная технология – и поклялся применять ее в любой схеме разбивки создаваемых мной дисков (по крайней мере, пока Btrfs не станет основной).

Сначала мы расскажем о том, что такое LVM и зачем это нужно. Затем возьмем экран установки в Fedora и покажем, как создать схему разбиения на разделы с нуля с помощью LVM. Мы также обратимся к командной строке и покажем вам инструменты, задействующие расширенные возможности LVM. Нужно изменить размер тома? Без проблем. Хотите добавить новый жесткий диск или удалить старый? У нас и это есть. Мы даже намерены показать вам, как сделать «снимок» [snapshot] и создать полноценную резервную копию системы в LVM. Мы много чего осветим!

Linux-хранилище 101

Все файлы и папки, принадлежащие системе Linux, и все программы, файлы настройки и файлы пользователя живут в древе каталогов. На вершине этого древа – корневая папка (отмечена /), а все остальное лежит под ней.

Конечно, все это хранится на физическом диске – жестком диске – а информация о том, где хранятся файлы на диске, хранится в так называемой файловой системе.

У вас могло сложиться представление о жестком диске как о едином пространстве для хранения, содержащего корневую папку и все, что ниже ее уровня; но такое бывает редко. В большинстве дистрибутивов предлагается «разбить» жесткий диск, разделив его на части, которые ОС рассматривает как физически уникальные диски.

Важным папкам также выделяется по отдельному разделу, каждый со своей файловой системой – многие дистрибутивы создают разделы для /, /home и /usr. Обратите внимание, что, поскольку разделы воспринимаются как дополнительные жесткие диски, они могут так же легко быть отдельными физическими дисками, как и разделами.

Вполне допустимо, чтобы все древо каталогов находилось в одном разделе, но по многим веским причинам так делают редко. Для домашних пользователей это чаще всего означает, что можно обновить или изменить дистрибутив без необходимости резервного копирования и восстановления содержимого вашей домашней папки, так как заменяются только разделы, содержащие системные файлы.

Для бизнес-пользователей это означает, что целые файловые системы можно предназначить только для чтения, обеспечив дополнительный уровень безопасности и стабильности и упростив резервное копирование. Более того, так как важные папки хранятся на различных частях диска, повреждение его части не влияет на остальной диск.

Розовые очки?

Все это кажется хорошо продуманной системой, но и здесь не без проблем. Больше всего удручает то, что это очень негибкая система; размер разделов трудно изменить, как путем создания новых жестких дисков, так и сжатия других разделов.

Представьте себе, например, что у вас есть жесткий диск на 80 ГБ, который вы решили разбить на разделы. Вы хотите как можно больше места выделить под /home, потому что у вас большая музыкальная коллекция. Таким образом, вы, прежде всего, решаете, какой минимальный объем пространства нужен для раздела /, и отводите оставшееся для /home. В конце концов / получаете 3 ГБ, а /home – 77 ГБ.



» Жесткие диски в Linux – это не просто большой блок с местом для хранения: это также MBR, разделы, файлы, папки и индексы.

ХРАНИЛИЩЕМ

Характеристики файловой системы

Мы уже упоминали тот факт, что LVM вполне может изменить размеры разделов без их размонтирования. Чтобы это работало, нужно использовать файловую систему, также поддерживающую эту функцию, причем поддержка бывает разной степени. Чтобы упростить вам жизнь, мы составили эту удобную таблицу, где отображено, какие файловые системы поддерживают эту функцию и насколько.

	Сжатие онлайн	Увеличение онлайн	Сжатие оффлайн	Увеличение оффлайн
ext3	✗	✓	✓	✓
ext4	✗	✓	✓	✓
btrfs	✓	✓	✗	✗
xfs	✗	✓	✗	✗
jfs	✗	✓	✗	✗
reiserfs	✗	✗	✓	✓

Все идет прекрасно, пока вы не вздумаете попробовать KDE или другой огромный пакет программного обеспечения и в разделе / не хватит места. Как назло, ваша музыка заняла только 70 ГБ, и теперь 7 ГБ пропадают даром.

Ну да, можно просто сделать резервную копию данных, переустановить дистрибутив и все восстановить – но это пустая трата времени. Разве не проще было бы добавить еще один жесткий диск и использовать его для раздела /, или просто и безопасно уменьшить размер /home на 5 ГБ и отдать освободившееся пространство для /?

Все учитывается

LVM делает все это возможным, то есть вам больше нечем страдать, что изначально вы промахнулись с планированием. С LVM можно динамически добавлять больше пространства старым разделам, растягивать разделы на несколько дисков и перераспределять ранее назначенное дисковое пространство. В зависимости от особенностей вашей файловой системы, многие из этих задач можно осуществить даже когда раздел используется.

LVM делает это, выступая, по сути, посредником между файловой системой и физическими дисками. Такое стало возможным благодаря разделению физических дисков на так называемые физические экстенды – куски, гораздо меньшие, чем разделы; их легче перемещать по диску, и над ними легче выполнять другие операции.

Более того, не имеет значения, на каком устройстве экстенды расположены – LVM разделит несколько дисков на такого же размера экстенды, а затем объединит их все для создания одного гигантского ресурса хранения – как виртуальный жесткий диск.

Хотя многое из того, что делает LVM, использует концепции, которые мы уже обсуждали, например, файловые системы и физические диски, здесь есть свой собственный жаргон, который вы должны освоить, прежде чем идти дальше.

» **Физические тома** Физические диски или разделы, которые LVM делит на физические экстенды.

» **Группы томов** Коллекции физических томов; гигантские пулы для хранения всех физических экстендов.

» **Логические тома** LVM создал виртуальный жесткий диск, группы томов, поэтому также потребуется создать виртуальные раз-

делы – или логические тома. Они могут быть отформатированы под файловую систему, используемую для хранения данных и монтируемую в основное древо каталогов, подобно обычным разделам.

Каждый из этих терминов формирует часть «стека» LVM, с физическими томами, покоящимися на дне, группами томов в середине и логическими томами на самом верху. Все они – необходимые элементы для работы LVM и должны создаваться в определенном порядке:

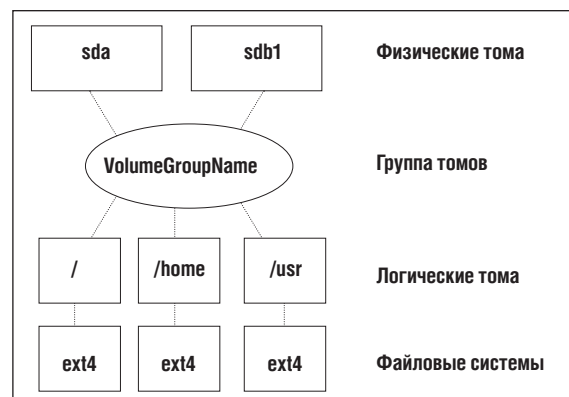
» Ваши **физические диски** или разделы должны быть зарегистрированы в LVM как доступные **физические тома**.

» Затем вы должны сказать LVM, какие **физические тома** вы хотите назначить каким **группам томов**.

» Наконец, вы должны сказать LVM, какие **логические тома** вам нужны в каждой **группе томов**, и какое соотношение места для хранения вы хотите назначить каждой из них.

Единственное ограничение, которое нужно иметь в виду при создании новой схемы разделов в LVM – она не может содержать загрузочный раздел, а значит, придется создать отдельный раздел /boot.

Когда вы все это сделаете, группа томов и ее логические тома будут доступны для монтирования и проверки в папке /dev, как обычный жесткий диск или раздел. Более того, их легко идентифицировать в /dev, так как они перечислены по названию, например /dev/Volume GroupName/LogicalVolumeName.

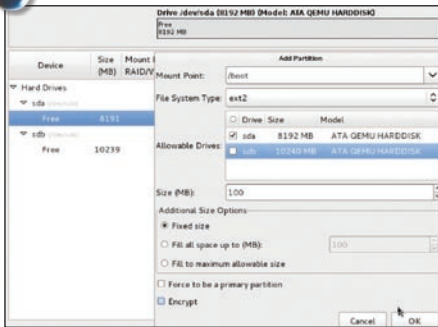


» Жаргон LVM затрудняет понимание, но на самом деле очень логичен, как видно на этой диаграмме. Для вашего вдохновения: <http://ltdp.org/HOWTO/LVM-HOWTO/anatomy.html>.

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

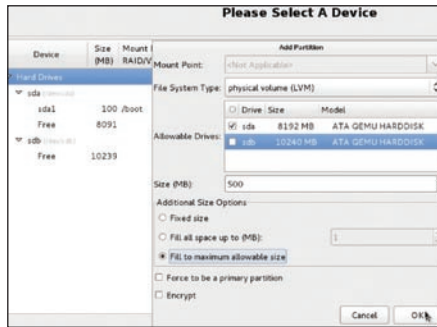


Шаг за шагом: Освоим экран разбиения диска в Fedora



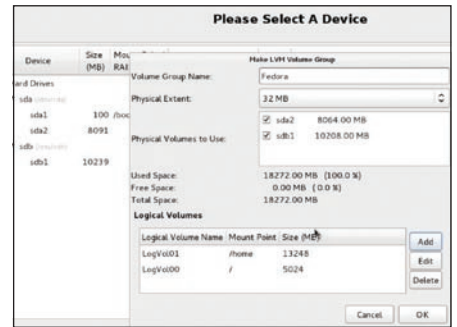
1 Создаем раздел

Начиная с пустого жесткого диска, нажмите кнопку Создать > Стандартное разбиение. Установите точку монтирования в /boot и выберите, на каком диске вы хотели бы хранить этот раздел. Он должен быть фиксированного размера, и только 100 МБ.



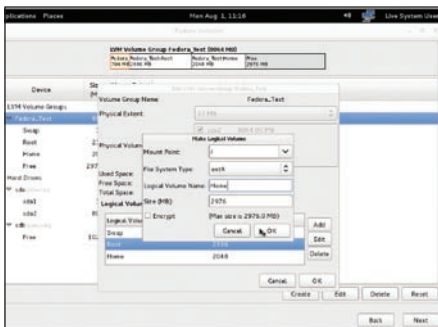
2 Создаем физический том

Затем зарегистрируйте остаток свободного пространства как физический том. Нажмите кнопку Создать > Физический том LVM, а затем установите Тип файловой системы для физического тома (LVM). Установите его в Заполнить до максимально допустимого размера.



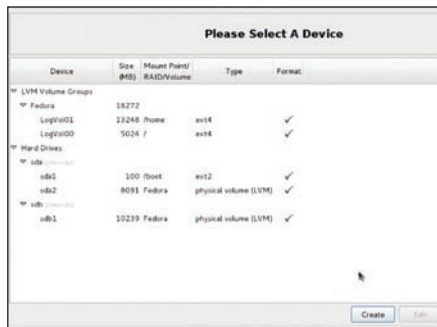
3 Создаем группу томов

Для создания группы томов и логических томов, нажмите кнопку Создать > Группа томов LVM. Задайте ей узнаваемое имя, а затем укажите, какие физические тома использовать. Размер физического экстенда можно оставить по умолчанию.



4 Создаем логические тома

И наконец, создайте логические тома, нажав кнопку Добавить в окне Группа томов. Укажите точку монтирования как /, /home или любую другую папку по вашему выбору, выберите файловую систему, дайте ей хорошее описательное имя и установите размер для нее.



5 Обзор изменений

Теперь просмотрите изменения, которые вы указали. Проверьте правильность дисков и разделов, зарегистрированных как физические тома, и убедитесь, что все они правильного размера и вы не удалите чего-нибудь ценного по ошибке.



6 Запишем изменения

Наконец, мастер установки предупредит вас, что он создаст новую схему дискового пространства, и любые данные на удаленных или отформатированных разделах будут потеряны. Просмотрев изменения, нажмите Записать изменения на диск.

Чтобы все было предельно ясно, мы создали пошаговую инструкцию по установке Fedora с использованием LVM, *вверху*.

Итак, вы познакомились с LVM и его терминологией, и смогли создать основанную на LVM схему разбиения диска с помощью пугающего мастера Fedora. Теперь обратим внимание на командную строку: начнем с показа, как создать новую LVM-систему, а затем рассмотрим, как ее изменить.

Первым шагом будет регистрация наших жестких дисков и разделов как имеющих в наличии физических томов, командой `pvcreate`:

```
pvcreate /dev/sda1
```

Она регистрирует первый раздел на первом жестком диске как доступный. Если вы хотите зарегистрировать весь второй жесткий диск как свободное пространство, запустите

```
pvcreate /dev/sdb
```

Обратите внимание, что, хотя LVM с удовольствием работает с целыми дисками, а не разделами, другие операционные системы могут отображать его как свободное пространство; будьте осторожны, чтобы случайно не удалить информацию.

Далее, создадим группу томов из зарегистрированных нами физических томов. Команда проста:

```
vgcreate VolumeGroupName /dev/sda1 /dev/sdb
```

VolumeGroupName здесь – очевидно – описательное имя для группы томов, а потом следуют физические тома, образующие пул дискового пространства. Если вы забыли, какие физические тома создали, запустите команду `pvscan` – вы получите список доступных дисков и разделов. `pvdisplay` можно использовать для получения подробного описания характеристик физического тома, таких как размер, расположение на диске, имя и т.д.

Затем создайте логические тома из свежесозданной группы томов:

```
lvcreate -L10G -nLogicalVolumeName VolumeGroupName
```

Ключ `-L` указывает размер нового логического тома, с суффиксами К для килобайт, М для мегабайт и G для гигабайт. `-n` задает имя новому логическому тому, а последний параметр определяет, из какой группы томов его создавать.

Вот все команды, имеющие непосредственное отношение к LVM, которые вы должны знать, и все они довольно просты.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Теперь создайте файловую систему на каждом логическом томе и позаботьтесь о ее монтировании. Создать файловую систему несложно, если вы намерены обратиться к стабильному и очень популярному семейству файловых систем ext. Вам нужно установить пакет **e2fsprogs**, который предоставит вам команду *mke2fs*. Укажите *mke2fs*, какую файловую систему ext использовать (4 – последняя и лучшая) и на каком логическом томе ее создавать:

```
mke2fs -t ext4 /dev/VolumeGroupName/LogicalVolumeName
```

Другие файловые системы имеют подобные инструменты, и ни на одной из них это не будет слишком сложной задачей. Google покажет, какие инструменты какой файловой системе понадобятся; страницы справки помогут в более тонких настройках.

После создания файловой системы на логическом томе вам останется ее подключить. Чтобы это происходило автоматически при каждой перезагрузке компьютера, добавьте логический том в файл */etc/fstab*.

Этот файл состоит из серии строк, по одной для каждой файловой системы, которая должна быть смонтирована. Каждая строка состоит из серии утверждений, разделенных пробелом или символом табуляции, с указанием раздела или устройства, точки монтирования, типа файловой системы, а также нескольких других параметров. Для раздела LVM запись будет выглядеть так:

```
/dev/VolumeGroupName/LogicalVolumeName /home ext4 defaults 0 1
```

Чтобы разобраться в деталях этой строки, начиная с **defaults**, не забудьте глянуть на страницу справки */etc/fstab*.

Добавление памяти

Пока мы только создали схему дискового пространства, функционально эквивалентную обычной системе с разделами. А теперь, когда все работает, пришло время разобраться, как использовать некоторые добавочные возможности LVM.

Сначала рассмотрим, как добавить пространства, в виде нового физического тома, в группу томов, а затем в логический том.

Как и для всех физических томов, первым делом вам нужно инициализировать его командой **pvcreate**. Сделав это, можно использовать команду **vgextend**, чтобы добавить новый физический том в пул группы томов:

```
vgextend VolumeGroupName /dev/sda2
```

Синтаксис довольно очевиден: **VolumeGroupName** – том группы для увеличения, а **/dev/sda2** – физический (уже инициализированный) том, за счет которого идет увеличение.

Теперь дополнительное пространство доступно, и вы можете расширить логические тома в группе томов по своему желанию. Для этого мы используем команду **lvextend**, синтаксис которой схож с командой **lvcreate** (в частности, ключ **-L**).

```
lvextend -L8G /dev/VolumeGroupName/LogicalVolumeName
```

Легко, не так ли? Это не увеличивает файловую систему, и прежде чем вы сможете воспользоваться пространством, следует увеличить и ее размер. Следующая команда расширит файловую систему ext4 до того же размера, что недавно измененный логический том:

```
resize2fs /dev/VolumeGroupName/LogicalVolumeName
```

Что здесь и вправду удивительно – то, что это можно сделать без перезагрузки системы. Вспомним наш пример с музыкой: когда место в нашей системе на 80 ГБ закончилось, мы могли бы добавить еще один жесткий диск и увеличить раздел */*, используя дополнительное пространство в пуле. С этим гораздо меньше хлопот, чем с выполнением полного восстановления системы.

Вас может заинтересовать, что нужно делать, если ваши 80 ГБ – это ноутбук, без какой-либо возможности подключить внешний жесткий диск. Можно ли будет сокращать размер */home*, отдавая это пространство для */*? Да, но с оговорками.

Важнейшая из них очевидна: нельзя сделать размер файловой системы меньше, чем данные, которые в ней содержатся. Иначе вы, разумеется, потеряете часть этих данных и рискуете испортить

Что такое Btrfs

Btrfs – это новая файловая система для Linux, призванная помочь крупным системам хранения данных, которые часто встречаются в центрах обработки данных и других промышленных установках. Она предоставляет мощный набор средств для управления и другие продвинутые возможности.

Btrfs также поддерживает так называемые «подтома [subvolumes]», отдельные внутренние корни файловой системы (кстати, именно это делает возможными снимки системы). Хотя они сильно напоминают логические тома LVM, это не совсем одно и то же, поскольку они являются не блочными устройствами, а монтируемыми папками внутри основного дерева файловой системы.

Несмотря на чисто технические различия, это очень схожие сервисы. Btrfs снимает большую часть потребности в LVM, и так как все это грузится в одну файловую систему без разделов или логических томов, администрирование достаточно упрощено. Более того, данная система безопаснее – благодаря таким функциям, как онлайн-проверка файловой системы, контрольные суммы, а также зеркало для файловых систем, которые отсутствуют в других файловых системах с LVM.

Все эти характеристики делают Btrfs очень привлекательной, и именно поэтому Fedora предлагает ее по умолчанию, вместо LVM, в Fedora 16. Мы сделаем урок по Btrfs очень скоро.

остальное. Итак, убедитесь, что в файловой системе достаточно свободного места, прежде чем сокращать ее размер.

Пункт второй – хотя LVM умеет сокращать логические тома, даже когда они смонтированы, большинство файловых систем эту возможность не поддерживают. Вместо этого, если вы вносите изменения в раздел типа */* или */home*, который должен быть смонтирован при работающей системе, вам необходимо перезагрузиться в режим работы с live CD или другой системы восстановления.

Помимо этих предостережений, для логического тома, содержащего файловую систему ext4, оставшаяся часть процедуры проста. Сначала запустите команду **resize2fs**, указав новый размер файловой системы, а затем запустите **lvreduce**, чтобы уменьшить логический том:

```
resize2fs /dev/VolumeGroupName/LogicalVolumeName 2G
```

```
lvreduce -L2G /dev/VolumeGroupName/LogicalVolumeName
```

Здесь **2G** в обеих командах просто указывает новый размер файловой системы и логического тома. Вот и все, что нужно сделать. Пускай это и страшно – сама процедура не слишком трудна.

Перемещение данных

Теперь рассмотрим, как удалить старый диск и заменить его на новый. Чтобы сделать это, не потеряв ничего, нужно скопировать все данные, содержащиеся на физическом томе, который будет удален, в другие части группы томов.

Это звучит сложно: в конце концов, как вообще можно выяснить, какие части группы томов LVM хранит на каком физическом томе? К счастью, в LVM есть еще одна удобная команда, чтобы сделать это невероятно просто:

```
pvmove /dev/sda
```

Это заставит все физические экстенды храниться на первом жестком диске, чтобы перераспределиться между другими физическими томами в группе томов – если хватит места. Если места недостаточно, сперва добавьте новый физический том, точно так же, как мы делали раньше, а затем запустите команду.

Достаточно ли места в группе томов, можно проверить, сравнивая выход **pvdisplay** с **vgdisplay**.

Осталось только удалить старый жесткий диск из группы томов, командой **vgreduce**:

```
vgreduce VolumeGroupName /dev/sda
```

Она удалит физический том */dev/sda* из группы томов **VolumeGroupName**. На этом этапе вы можете удалить жесткий диск – сжечь его, разбить его... короче, позаботиться, чтобы никто другой не заполучил в свои руки ваши данные.

Тут уроку и конец. Надеюсь, мы открыли вам глаза на потенциал LVM, и вы тоже приметесь использовать ее в своих будущих системах. **LXF**



MediaWiki. Сам

Это свободное ПО – база 65% публичных wiki. Освойте основы – и вашей креативности не будет преград! **Бен Эверард** вам все разъяснит.



Наш эксперт

Бен Эверард

Бросил работу ИТ-консультанта, чтобы провести два года в Танзании за установкой в школах систем на базе Ubuntu. Теперь он проводит свое время в писательстве про свободное ПО и свои путешествия.

Wiki, кажется, оплели всю Всемирную паутину, в том числе Wikipedia – седьмой из наиболее посещаемых сайтов – и руководства пользователя большинства дистрибутивов Linux. По сути, wiki – это просто система быстрого создания и редактирования web-страниц, которые можно использовать для чего угодно: Интернет-энциклопедии, руководства или списка покупок.

На нашем уроке мы пройдем процесс создания вашей собственной wiki и покажем вам, как создать несколько страниц, чтобы помочь вам стать организованнее. Эти страницы – просто примеры: когда wiki у вас заработает и вы поймете, как редактировать страницы, вам откроются безграничные возможности.

Все в wiki зиждется на редактируемости, и эта система позволит редактировать ваши страницы с настольного компьютера, ноутбука, смартфона или любого другого устройства, подключенного к локальной сети. Если соединение позволяет, вы можете даже открыть свою wiki для всего Интернета.

Подготовка

Программное обеспечение *MediaWiki* работает на сервере LAMP. Если он у вас уже настроен, вы готовы к работе. Если нет – не волнуйтесь: настроить его несложно даже на обычном настольном ПК. L означает Linux, и тут сгодится любой дистрибутив, хотя если вы предназначаете его для данной задачи, серверный (например, Ubuntu Server) будет лучшим выбором. А, М и Р обозначают *Apache* (web-сервер), *MySQL* (базы данных) и *PHP* (язык скриптов). Процесс установки несколько различен в разных дистрибутивах.

» **Debian** и производные (**Ubuntu, Mint** и т.д.) Откройте менеджер пакетов *Synaptic* (если вы устояли против Unity, его можно найти в разделе Система > Администрирование) и выберите пакеты **Apache2, PHP5, MySQL-server** и **php5-MySQL**. Нажмите кнопку Применить, а при появлении запроса введите пароль администратора для *MySQL* – он вам потребуется при настройке wiki.

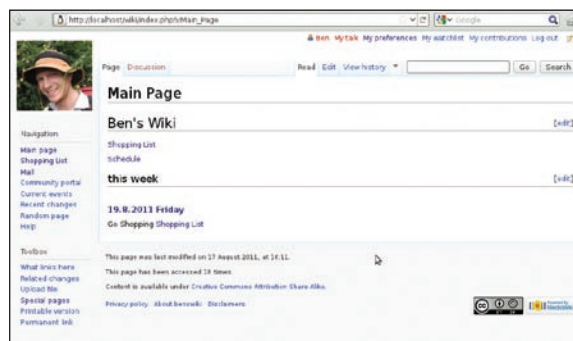
» **SUSE** Установите пакет **patterns-openSUSE-lamp_server**, затем откройте **YAST** > Система > Системные службы (Уровень загрузки). Переключите радиокнопку на Эксперт. И для *Apache*, и для *MySQL*, выберите соответствующую строку и отметьте флажками три и пять, затем из выпадающего меню запустите сервис. Папка для web-сервера **/srv/www/htdocs**, нужно заменить на это **/var/www** в командах, приведенных в данном учебнике.

» **Fedora** и **Centos** Выберите Приложения > Системные утилиты > Установка и удаление программ и установите пакеты **httpd, mysql-server, php, php-mysql** и **ntsysv**. После этого скомандуйте

```
sudo ntsysv
```

Прокрутите вниз до **httpd** и **mysqld**, отметьте их и нажмите кнопку ОК. Установятся web-сервер и сервер базы данных для запуска при загрузке системы. Теперь запустите серверы – это можно сделать из командной строки или через перезагрузку. Web-сервер использует папку **/var/www/html**, и ее нужно использовать вместо **/var/www** в этом учебнике.

» **Mandriva Linux** Понадобится открыть Установка и удаление программ и установить пакет **task-lamp**. Соответствующие серверы установятся и настроятся на запуск при загрузке, но не запустят-



» На *MediaWiki* работает *Wikipedia* – ее англоязычная версия содержит более 3 миллионов страниц.

ся. Теперь вы можете запускать серверы вручную из командной строки или перезагрузить компьютер.

Вам нужно будет заменить **/var/www** на **/var/www/html** в остальной части этого учебника, а также придется заменить **sudo** в командах на **su -c**, затем заключить команду в одиночные кавычки. Например, замените:

```
sudo tar zxvf mediawiki-1.17.0.tar.gz -C /var/www/
```

```
на su -c 'tar zxvf mediawiki-1.17.0.tar.gz -C /var/www/html'
```

Установив все это, откройте web-браузер и введите адрес **http://localhost/**. Если вы получили сообщение об ошибке, вам придется вернуться назад и исправить *Apache*. Если web-страница сообщит, что *Apache* работает, то вы готовы к установке *MediaWiki*. Запустите **mysql_secure_installation** из командной строки и следуйте инструкциям, чтобы обезопасить свою базу данных.

Скачайте архив с <http://www.mediawiki.org/wiki/Download>. Вы получите файл с именем **MediaWiki-1.17.0.tar.gz** (или более новую версию). Откройте терминал, перейдите в этот файл и распакуйте содержимое в web-папку, командой

```
sudo tar zxvf mediawiki-1.17.0.tar.gz -C
```

```
/var/www/ – папка по умолчанию для новых установок Apache на производных Debian, но если ваш сервер настроен иначе, надо будет изменить эту папку (см. выше, про установку LAMP в вашем дистрибутиве). Создастся новая папка /var/www/mediawiki-1.17.0, содержащая все скрипты и данные для wiki. Можно дать ей имя покороче – wiki, с помощью
```

```
sudo mv /var/www/mediawiki-1.17.0 /var/www/wiki
```

Откройте web-браузер, укажите в нем **http://localhost/wiki**, и вы должны получить страницу с сообщением «LocalSettings.php не найден». Нажмите на Настройка Wiki.

Выберите языки для wiki и нажмите Продолжить. Система проверит окружение, чтобы убедиться, что все необходимое ПО установлено и работает. По ходу вы получите несколько предупреждений, но если вы дождетесь ярко-зеленого сообщения «Условия проверены, вы можете перейти к установке *MediaWiki*», то все в порядке. Сообщение об ошибке красного цвета означает, что вам придется исправить соответствующие приложения, а потом уж двигаться дальше.

Скорая помощь

К каждой странице wiki привязана страничка обсуждения – на нее можно попасть по щелчку на вкладке Discussion в верхнем левом углу. Подробнее см. на http://localhost/wiki/index.php/Help:Talk_pages.

себе Википедия

Нажмите на Продолжить, чтобы принять GPL и создать свою wiki. На следующем экране нужно ввести детали для подключения wiki к MySQL. Если у вас нет никаких особенных требований, можете оставить все параметры по умолчанию, кроме тех, что в нижней части окна: там вводится пароль для базы данных, который вы создали при запуске `mysql_secure_server`.

Следующая страница позволит выбрать несколько вариантов базы данных; самый важный из них – верхний, Использовать одну учетную запись для установки. Если снять это условие, возникнет дополнительный набор флажков для создания базы данных. Введите имя пользователя и пароль, а затем установите флажок Создать учетную запись если она еще не существует. Это предотвратит доступ wiki к вашей базе данных с правами администратора, что дает дополнительную защиту.

Нажав на кнопку Продолжить, вы можете ввести имя wiki (я назвал свою Benswiki, но имя может быть любым) и детали для администратора (это вы). Нажмите кнопку Далее еще раз, чтобы получить последнюю страницу опций.

Укажите здесь права пользователя как Традиционная wiki, то есть разрешите всем редактировать страницы, независимо от наличия учетной записи. По умолчанию, авторские права и лицензии означены как Подпись без лицензии, но если вы хотите обеспечить, чтобы ваша работа осталась свободной, измените это на Creative Commons Attribution Share Alike. Остальное сгодится по умолчанию, кроме Включить выгрузку файлов – это нужно указать; затем нажмите кнопки Далее, Продолжить и снова Продолжить для установки wiki, и еще раз Продолжить, чтобы скачать файл `LocalSettings.php`. Он содержит все параметры настройки, и его надо скопировать в должную папку – откройте терминал, перейдите туда, куда скачался файл, и введите

```
sudo cp LocalSettings.php /var/www/wiki
```

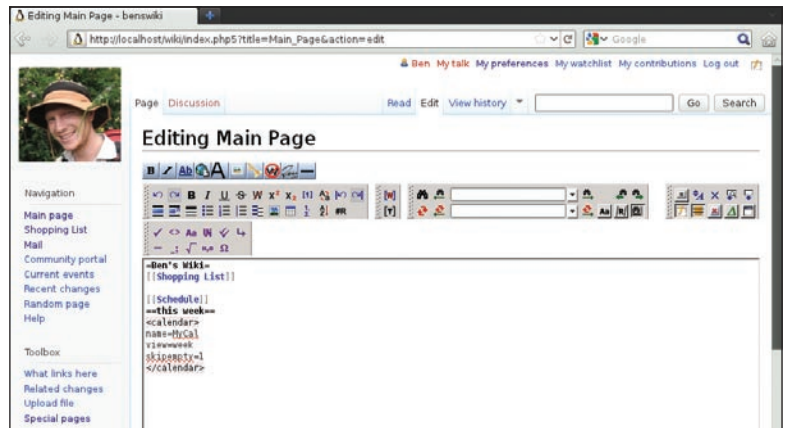
После этого можете нажать на Войдите в вашу Wiki, или наберите `http://localhost/wiki` в адресной строке браузера, чтобы открыть главную страницу сайта. Теперь у вас есть своя собственная wiki в рабочем состоянии, и можно начать собирать на ней информацию. Если ваш браузер показывает адрес `localhost/wiki/index.php5/Main_Page`, а не `localhost/wiki/index.php/Main_Page`, нужно вставить 5 в URL-адреса, приведенные в оставшейся части нашего урока.

Все настройки для wiki хранятся в файле `LocalSettings.php`. Разметка страницы по умолчанию неплоха, но мы персонализируем ее картинкой. В левом верхнем углу всех страниц помещен логотип wiki, и его можно заменить на любой рисунок по вашему выбору. Так как я создаю личный wiki, я помещу там свой портрет.

Размер картинки – 135×135 пикселей; более крупные иллюстрации будут обрезаны до него. Выбрав изображение (я взял фото с именем `my_face.jpg`), откройте терминал, перейдите в папку с изображением и скопируйте его в `wiki/skins/common/images` командой

```
sudo cp my_face.jpg /var/www/wiki/skins/common/images/
```

Затем отредактируйте путь к рисунку в файле `LocalSettings.php`. Чтобы открыть файл в `nano` с правами суперпользователя, введите в терминале



```
sudo nano /var/www/wiki/LocalSettings.php
```

В текстовом редакторе найдите строку, начинающуюся с `$wgLogo` (она должна быть примерно на уровне трети файла). Измените ее так, чтобы она указывала на файл:

```
$wgLogo = "$wgStylePath/common/images/my_face.jpg";
```

Сохраните файл (Ctrl-O в `nano`) и обновите страницу в браузере. Ваше изображение должно теперь заменить логотип *MediaWiki*.

Добавление содержимого

Главная страница вашей wiki, как и все страницы, которые вы впоследствии создадите, имеет сверху два набора вкладок. В левом – два варианта: Страница и Обсуждение. В правом – три: Читать, Редактировать и Просматривать историю. Их можно использовать для отображения страниц позади страницы. Первым делом мы воспользуемся Edit – нажмите на эту вкладку, чтобы открыть форму редактирования.

Основная часть страницы Редактировать – текстовое поле, содержащее код для главной страницы на языке разметки wiki. Сначала удалите этот текст, потому что это просто пример, а затем дайте странице новый заголовок:

» Текстовый редактор *WikiEd* добавит функции и подчеркивание синтаксиса на странице редактирования.

Доступ с другого компьютера

Вы можете получить доступ к wiki с любого другого компьютера в той же сети, используя ссылку `http://w.x.y.z/wiki`, где `w.x.y.z` – IP-адрес машины. Узнать свой адрес вам поможет команда `ifconfig` (возможно, потребуется использовать `sudo`).

Выведется по блоку текста для каждого сетевого интерфейса на компьютере. Найдите блок, относящийся к сети, из которой вы хотите получить к нему доступ, и вы увидите раздел в виде `inet addr:w.x.y.z - w.x.y.z` является IP-адресом этого компьютера. Если этот адрес меняется при каждой

загрузке компьютера, надо назначить ему статический IP-адрес на маршрутизаторе.

Если ваш провайдер позволяет, можно получить доступ к wiki из Интернет. Для этого настройте проброс портов (иногда виртуальных серверов) на маршрутизаторе для перенаправления web-трафика на вашу wiki. Более подробные инструкции вы найдете в руководстве пользователя маршрутизатора. После этого можно использовать службу вроде DynDNS (`http://dyn.com/dns/dyndns-free/`) для получения имени хоста в форме `youname.dyndns.org`.

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



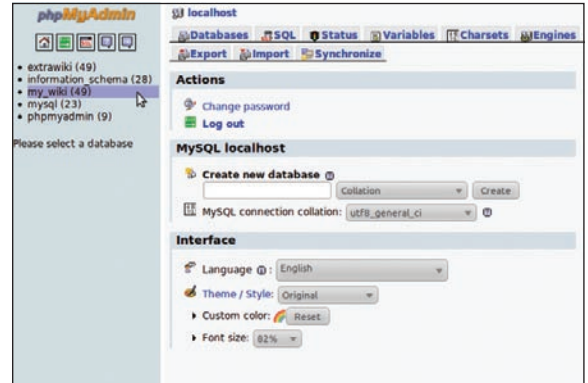
Шаг за шагом: Резервная копия системы

Большая часть данных для wiki хранится в базе данных MySQL, и ее резервное копирование немного сложнее, чем просто копирование файлов в надежное место. Но не волнуйтесь, хорошие люди в PhpMyAdmin создали инструмент, который все упрощает.

Чтобы воспользоваться руководством, установите *PhpMyAdmin*. Для большинства пользователей это означает установку пакета **phpmyadmin** (Пользователям SUSE может потребоваться скачать его с software.opensuse.org).

Скорая помощь

Кроме вкладки Редактировать, каждая страница содержит вкладку История – ее можно использовать для отката изменений.

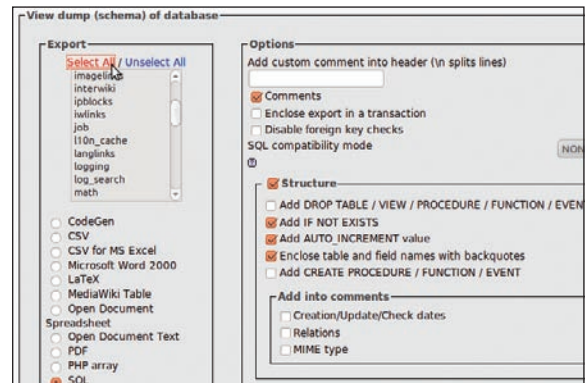
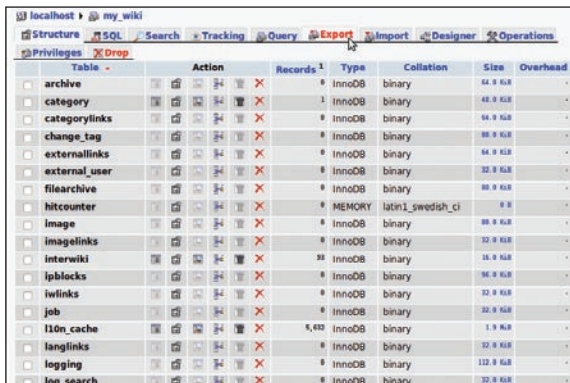


1 Откройте браузер

Введите <http://localhost/phpmyadmin> и войдите в систему с именем пользователя и паролем для базы данных MySQL (как root). Это также может быть <http://localhost/phpMyAdmin>.

2 Выберите базу данных

Выберите базу данных, соответствующую wiki, из списка баз данных, приведенного слева.



3 Экспорт

Нажмите на вкладку Экспорт. На этой странице перечислены всякие таблицы, которые можно выбрать вручную. Но мы лучше экспортируем все это целиком.

4 Скачать файл SQL

Нажмите кнопку Выбрать все, а затем Перейти. Загрузится SQL-файл. Сохраните его вместе с копией вашего файла **LocalSettings.php** в безопасном месте. Для воссоздания базы данных используйте вкладку Импорт.

```
= Ben's Wiki =
  Пусть эта главная страница содержит ссылки на две другие
  страницы, содержащие регулярно обновляемую информацию:
  [[Список покупок]]
  [[Расписание]]
```

Обратите внимание на пустую строку – *MediaWiki* игнорирует одиночные «переводы каретки», и пустые строки нужны для разбивки текста на отдельные строки. Двойные квадратные скобки говорят системе, что это ссылки на новые страницы. Нажмите кнопку Сохранить страницу, чтобы запомнить изменения, и перейдите к новой главной странице. Вы увидите свой заголовок и две ссылки красного цвета. Красный цвет указывает, что они ссылаются на страницы, которые еще не существуют. Нажмите на ссылку

ку **Список покупок** для создания страницы и переходите к редактированию раздела, чтобы добавить свои элементы:

```
# молоко
# яйца
# хлеб
```

велит системе форматировать это как нумерованный список. Благодаря этому нам не нужно вводить пустые строки между элементами. Нажав на Сохранить страницу, мы перейдем на страницу, где вы увидите свой список покупок.

Нажмите на главную страницу под Навигация слева, чтобы вернуться на главную страницу. Ссылка **Список покупок** должна стать фиолетовой, показав, что страница теперь существует.

В wiki все, включая боковую панель, хранится в виде страниц, которые можно редактировать. Укажите в браузере <http://>

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

`localhost/wiki/index.php/MediaWiki:Sidebar` – откроется страница содержимого этой панели. Это особая страница, и прежде чем ее редактировать, нужно войти в систему – сделайте это, используя ссылку в правом верхнем углу страницы. Войдя в систему, переберитесь на вкладку Правка и добавьте строку:

```
** Список покупок | Список покупок
```

Две звездочки – это разметка wiki, которая говорит системе, что эта линия является частью маркированного списка, с одним уровнем. Вы можете поместить здесь ссылки на внешние сайты, например, на web-почту. Добавьте

```
** http://mail.yahoo.com!Mail
```

Подставьте сюда сайт своей web-почты. Сохранив страницу, вы увидите два новых элемента на боковой панели.

Свежая установка *MediaWiki* снабжена всем программным обеспечением, но ни одного файла помощи в ней нет. Мы можем экспортировать их с web-сайта организации. Укажите в браузере www.mediawiki.org и перейдите к Панели инструментов (на боковой панели) > Спецстраницы. Перейдите на Инструменты страницы > Экспорт страниц и в поле Добавить страницы из категории введите Справка и нажмите кнопку Добавить. Появится список всех страниц помощи. Нажмите кнопку Экспорт, чтобы скачать их на компьютер в виде файла XML.

Откройте свою wiki в браузере и нажмите на кнопку Специальные страницы > Инструменты страницы > Импорт Страниц. Выберите только что скачанный файл и нажмите Загрузить файл. Справочные страницы добавятся в wiki, и вы можете получить к ним доступ, нажав Помощь в меню Навигации. Синтаксис разметки wiki можно найти в разделе Помощь > Редактирование > Форматирование.

А не здорово ли было бы модернизировать текстовый редактор, чтобы он помогал нам с форматированием? К счастью, мы можем добавить функциональность к нашей wiki с помощью расширений. Мы добавим две функции: одну – для обеспечения календаря, а другую – чтобы обзавестись текстовым редактором получше (этот продвинутый редактор работает с *Firefox*, *Chrome*, *Epiphany* и *SeaMonkey*, но не *Konqueror*).

Скачайте файл **wikicalendar-1.16.tar.gz** с <https://github.com/christofdamian/wikicalendar> и распакуйте его содержимое в папку с расширениями:

```
sudo tar xzvf wikicalendar-1.16.tar.gz -C /var/www/wiki/extensions
```

Затем откройте **LocalSettings.php** в текстовом редакторе (как вы делали при настройке wiki) и добавьте строку

```
require_once("$IP/extensions/wikicalendar-1.16/calendar/Calendar.php");
```

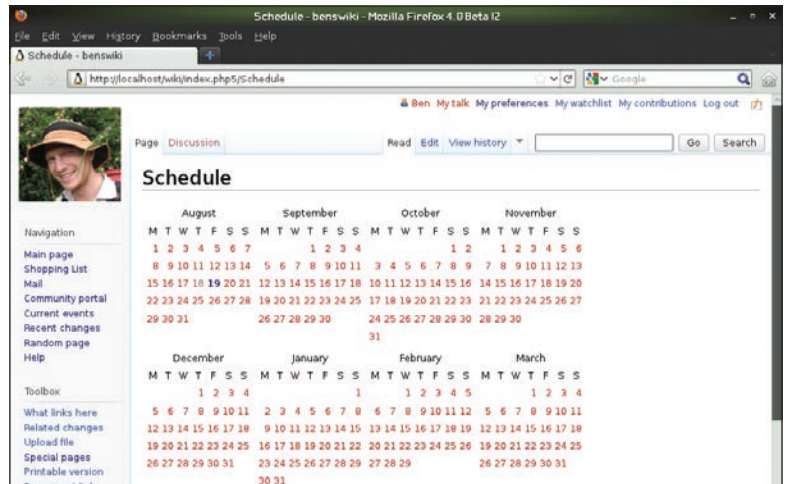
Откройте браузер и перейдите на страницу <http://localhost/wiki/index.php/MediaWiki:Common.js>; зайдите на вкладку Создать (если вы вышли из системы, необходимо снова войти) и введите текст:

```
// install the wikEd text editor
document.write('<script type="text/javascript" src="'
+ 'http://en.wikipedia.org/w/index.php?title=User:Cacycle/wikEd.js'
+ '&action=raw&ctype=text/javascript"></' + 'script>');
```

Затем нажмите Сохранить. Перейдя на главную страницу, нажмите на кнопку Расписание. Страница расписания создастся и откроется для редактирования. Вы обнаружите, что редактор теперь имеет подсветку синтаксиса, а также ряд кнопок для форматирования (если вы пользуетесь Wikipedia, вы получите тот же текстовый редактор, отметив *WikEd* на странице Настройки).

Воспользуемся расширением календаря, чтобы создать нашу страницу расписания. Введите

```
<calendar>
name=MyCal
view=months
months=12
</calendar>
```



Нажмите Сохранить для просмотра страницы. Вы увидите календарь на ближайшие 12 месяцев. Каждый день снабжен ссылкой красного цвета – она указывает, что это страницы, которые можно создавать и изменять, как любые другие страницы. Нажмите на красную ссылку на завтра и введите в редактировании страницы:

```
Поход по магазинам [[Список покупок]]
```

Нажмите Сохранить. Вы увидите, что **Список покупок** теперь ведет на ранее созданную страницу. При разметке wiki о путях можно не думать: все ссылки делаются по именам.

Мы получили календарь, но было бы полезно видеть все наши задачи на текущую неделю. Вернитесь на главную страницу, перейдите на вкладку Правка и введите следующие ниже строки после уже добавленного содержания:

```
==This Week==
<calendar>
name=MyCal
view=week
skipempty=1
</calendar>
```

Нажав Сохранить, вы увидите список задач на неделю (пока это просто напоминание сходить в магазин). Если вы решите идти в магазин не завтра, а послезавтра, можете переместить страницу, чтобы она указывала на другую дату. Нажмите на ссылку на Завтра для отображения в нем события. В верхней правой части страницы вы увидите стрелку вниз, между Просмотр истории и Поиск. Если вы поместите на нее курсор мыши, отобразится меню действий. Выберите Переместить. На каждую страницу события ссылка идет по названию, связанному с названием календаря и датой. Просто введите новую дату события в окно К новому названию, а затем снимите флажок Оставить перенаправление и нажмите Переместить страницу.

Желая избавиться от события, можно просто открыть вкладку Редактировать страницы и удалить текст, но сама страница останется, и календарь будет показывать событие (хотя и пустое). Чтобы убрать страницу полностью, нужно удалить ее. Для этого откройте страницу, перейдите на стрелку вниз рядом с Просмотром истории, выберите Удалить и нажмите кнопку Удалить страницу.

Случайно удаленную страницу можно восстановить, зайдя в Панель инструментов > Спецстраницы > Инструменты страницы > Просмотр удаленных страниц: выполните поиск вашей страницы или просмотрите журнал удаленных страниц. Обнаружив желаемую страницу, нажмите на Просмотр/Восстановление, а затем на Восстановить.

Вот и все, что нужно сделать. Теперь вы умеете создавать и редактировать страницы, и можете сделать свою wiki настолько сложной или простой, как вам угодно. **LXF**

Этот календарь – одно из сотен расширений, доступных на mediawiki.org.

КОМАНДЫ: GNU/

Тихон Тарнавский составляет скрипт, а скрипт составляет словарную тетрадку – вот так линуксоиды изучают языки.

Наш эксперт

Тихон Тарнавский работает в Linux-консоли больше восьми лет и точно знает, как сделать эту работу удобной.

Продолжаем начатое в предыдущей статье. Напомним: начали мы «работать со словарем» и успели выделить из словарной статьи интересующую нас информацию.

Теперь перейдем к тому, что же с этой информацией делать дальше. Как и раньше, будем записывать нужные команды в скрипт. Напомним, что самодельные словарики наподобие того, который мы будем составлять, на уроках иностранных языков часто называли словарными тетрадками. Поэтому и скрипт предлагаю назвать *tetradka*. А расположить его в уже привычном нам каталоге `~/bin/`.

На чем мы остановились?

Начнем с того, что у нас уже сделано в прошлый раз. Выделенное слово нам наверняка пригодится в скрипте не один раз, поэтому лучше сразу сохранить его в переменной:

```
#!/bin/bash
word="$(xclip -o)"
```

А в другой переменной сохраним полученный в предыдущей статье текст переводов

```
text="$(dict "$word" | tail -n +6 | sed -r 's/^[0-9]+[)]/)/'; s/^[^
[IVX]+/ /' |
tr '\n' ' ' | tr ' ' '\n' | sed -r 's/; * [a-Z]+.*$/ /' | tr ' ' '\n' |
tr -s ' ' | sed -r 's/^\s*\s*\s*/ /' | sed -r 's/^\s*\s*\s*/ /' | sed -r 's/^\s*\s*\s*/ /' |
sed 's/^[^)]*) */; s/ *[-:][a-Z :-]*$/; s/([^\s]*)[a-Z][^\s]*) */' |
egrep -v '^$|^([a-я])$)'"
```

Как вы, возможно, помните из предыдущей статьи, текст у нас здесь многострочный. Да, переменные командной оболочки могут хранить и многострочные «значения». Главное при этом – не забывать о кавычках как при назначении таких переменных, так и при их использовании. Напомним, что имена переменных внутри кавычек заменяются значениями этих переменных, а пробелы и прочие «специальные» для командной оболочки символы не обрабатываются.

Для начала обработаем пару нестандартных ситуаций. Во-первых, если ничего не выделено, просто молча прекратим работу. Проверить мы это можем уже известной нам командой `test` или `'[`. У нее есть опция `-z`, которая проверяет, что переданная ей строка пуста:

```
[-z "$word"] && exit 0
```

А можно сделать и наоборот – проверить, что в строке что-то есть:

```
[-n "$word"] || exit 0
```

Или даже так:

```
["$word"] || exit 0
```

Во-вторых, проверим, что произойдет, если слово не обнаружится в словаре:

```
$ dict qwe
No definitions found for "qwe", perhaps you mean:
mueller7: awe ewe owe we
```

Логично предположить, что это сообщение выводится в стандартный поток ошибок. Проверим:

```
$ dict qwe 2>/dev/null
$
```

Такой случай команда `dict` не считает ошибкой, поэтому ее сама в качестве условия использовать не получится. Но у нас есть другое простое условие: как видно в последнем примере, в этом случае на стандартный вывод вообще ничего не поступит, то есть переменная `text` останется пустой. Пустые строки мы проверять уже умеем. Но на этот раз не будем выходить молча, а отобразим предварительно окно с сообщением о том, что перевод найти не удалось (рис. 1):

```
["$text"] || {
zenity --error --text="Слово '$word' не найдено."
exit 1
}
```

Все вместе будет выглядеть так:

```
#!/bin/bash
word="$(xclip -o)"
["$word"] || exit 0
text="$(dict "$word" 2>/dev/null | tail -n +6 |
sed -r 's/^[0-9]+[)]/)/'; s/^[^
[IVX]+/ /' |
tr '\n' ' ' | tr ' ' '\n' | sed -r 's/; * [a-Z]+.*$/ /' | tr ' ' '\n' |
tr -s ' ' | sed -r 's/^\s*\s*\s*/ /' | sed -r 's/^\s*\s*\s*/ /' | sed -r 's/^\s*\s*\s*/ /' |
sed 's/^[^)]*) */; s/ *[-:][a-Z :-]*$/; s/([^\s]*)[a-Z][^\s]*) */' |
egrep -v '^$|^([a-я])$)'
["$text"] || {
zenity --error --text="Слово '$word' не найдено."
exit 1
}
```

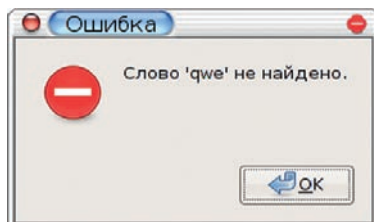
«Выбирайте, господа!»

Теперь нужно полученный текст показать в каком-нибудь окошке с возможностью выбора из имеющихся вариантов. Здесь нам снова поможет утилита `zenity`: у нее есть опция `checklist`, выводящая список с галочками для выбора. У аргумента этой опции достаточно своеобразный синтаксис: `"TRUEIFASLE текст1 TRUEIFALSE текст2 ..."`

Поэтому, прежде чем передавать ей текст, его нужно сначала обработать – привести к нужному виду. Запись `TRUEIFALSE` в примере означает, что в этом месте может стоять либо `TRUE`, либо `FALSE`. Первое обозначение подразумевает отмеченную «птичку», второе – не отмеченную. Нам логично выбрать предоставить человеку (то есть самому себе), то есть по умолчанию не выбирать ничего. Другими словами, в каждом пункте у нас будет `FALSE`. Теперь вспомним, как выглядит наша переменная `text` на данный момент – добавим в конец скрипта строчку `echo "$text"`, выделим нужное слово и запустим скрипт:

```
$ tetradka
свободный, вольный
находящийся на свободе
независимый
добровольный, без принуждения
незанятый, свободный
```

» Рис. 1. Ошибка: слово не найдено.



Linux и смекалка



легкий, грациозный
 неограниченный, не стесненный правилами, обычаями и т. п.
 щедрый
 обильный
 бесплатный, даровой
 освобожденный от оплаты
 открытый, доступный
 неприкрепленный, незакрепленный
 лишенный
 свободный
 free currency необратимая валюта, валюта, не имеющая обеспечения
 свободно
 бесплатно
 освобождать
 выпускать на свободу

Этот текст нам нужно сделать вот таким:
 "FALSE 'первая строка текста' FALSE 'вторая строка текста' ..."

Апострофы, окружающие каждое значение слова, нужны для того, чтобы *zenity* не восприняла пробелы внутри этих значений как разделители полей. Достаточно просто понять, как такое преобразование с текстом сделать: нужно в начале каждой строки дописать **FALSE**, а в конце – один апостроф. И затем заменить все символы перевода строки пробелами. Все это мы уже умеем делать: добавление строк в начало и в конец с помощью *sed*, а замену символов перевода строки – с помощью *tr*:

```
echo "$text" | sed "s/^/FALSE '/; s/$/'/" | tr '\n' ' '
```

Именно такой конвейер мы подставим в командную строку *zenity* – конечно же, окружив его конструкцией подстановки вывода – "\$(...)".

Теперь разберемся с другими нужными нам опциями *zenity*. Во-первых, нужно задать тип окна и его заголовок. Нужный нам перечень с галочками – это частный случай списка, так что тип окна будет *list* (список). А в качестве заголовка логично будет выбрать слово, которое мы переводим:

```
zenity --list --text="$word"
```

Далее, в тех случаях, когда текста в переводе будет достаточно много, умолчательный размер этого окна окажется не слишком удобен, поэтому добавим еще изменение ширины (*width*) и высоты (*height*) окна:

```
zenity --list --text="$word" --width=600 --height=450
```

А теперь перейдем собственно к типу и содержимому списка. Сначала нужно задать тип списка – у нас это будет *checkboxlist*. Затем с помощью повторения опции *column* с аргументом задаются имена колонок. Колонок у нас будет всего две: в первой – галочки, во второй – варианты перевода:

```
zenity --list --text="$word" --width=600 --height=450 \
--checkboxlist --column=# --column=перевод
```

Напомню, что обратный слэш в конце строки «отменяет» символ перевода строки, который идет сразу после него. Другими словами, это достаточно распространенный способ разбивки длинных строк в скриптах для читабельности. Не забудьте только,

что перевод строки не заменяется пробелом, а именно «исчезает». Именно поэтому перед слэшем стоит пробел – если его не вставить, две опции склеятся вместе и *zenity* выдаст ошибку:

```
$ zenity --list --text="$word" --width=600
--height=450\
```

```
> --checkboxlist --column=# --column=перевод
```

Этот параметр недоступен. Используйте *--help* для просмотра всех возможных параметров.

Если после выполнения этой команды вы вызовете ее из истории стрелкой вверх, то наглядно увидите отсутствие пробела:

```
$ zenity --list --text="$word" --width=600 --height=450--checkboxlist
--column=# --column=перевод
```

Теперь добавляем список, который мы сформировали чуть раньше:

```
zenity --list --text="$word" --width=600 --height=450 \
--checkboxlist --column=# --column=перевод \
"(echo "$text" | sed "s/^/FALSE '/; s/$/'/" | tr '\n' ' ')"
```

Но если вы запустите полученную команду в таком виде, получите пустое окно списка с одной галочкой (рис. 2). В чем же дело? А в обработке кавычек и апострофов командной оболочкой. Ведь такая обработка проводится при запуске команды или конвейера всего один раз, причем перед запуском всех программ, входящих в конвейер. То есть апострофы, поставленные командой *sed*, обработаны не были: на момент обработки их еще там нет. Нетрудно понять, как выйти из этой ситуации: нужно приказать командной оболочке обработать команду второй раз. Для этого в оболочке есть специальная встроенная команда **eval**:

```
eval zenity --list --text="$word" --width=600 --height=450 \
--checkboxlist --column=# --column=перевод \
"(echo "$text" | sed "s/^/FALSE '/; s/$/'/" | tr '\n' ' ')"
```

Теперь результат будет таким, как и ожидалось (рис. 3). Запускаем скрипт заново, выбираем несколько вариантов и посмотрим, что выдаст *zenity* на стандартный вывод:

```
$ tetradka
свободный, вольный/находящийся на свободе
```

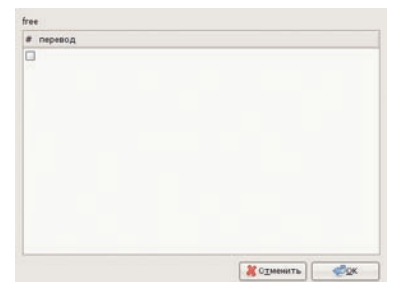
Как видите, выбранные варианты разделены вертикальной чертой. Заменяем ее переводом строки, после чего допишем в начало каждой строки переводимое слово:

```
eval zenity --list --text="$word" --width=600 --height=450 \
--checkboxlist --column=# --column=перевод \
"(echo "$text" | sed "s/^/FALSE '/; s/$/'/" | tr '\n' ' ')"
```

```
tr '\n' '\n' | sed "s/^/$word/"
```

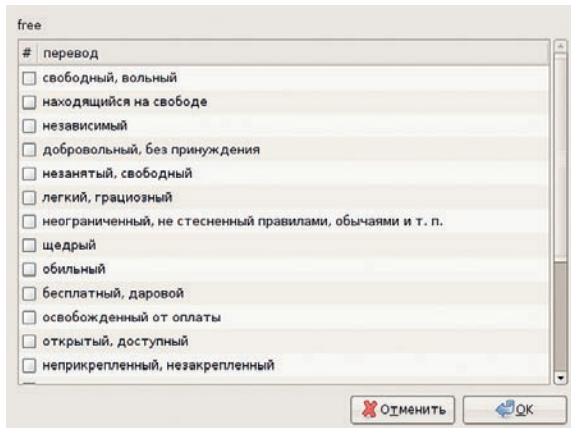
Очевидно, есть смысл записывать все это в некий файл. Только известное нам перенаправление вывода символом '*>*' здесь не подойдет. Ведь при нем файл каждый раз перезаписывается заново, а нам нужно добавлять новые слова с переводами в уже существующий. Для этого в командной оболочке есть другая, родственная

» Рис. 2. Без дополнительной обработки видим лишь пустой список.



» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

► Рис. 3. А теперь все правильно.



конструкция перенаправления – '>>'. Отличается такой сдвоенный знак от одиночного именно поведением в том случае, когда заданный файл уже существует: в этом варианте вывод команды не перезапишет файл, а добавится в конец. Файл, в который мы будем складывать слова с переводами, логично назвать **.tetradka** (исходя из имени скрипта) и разместить в домашнем каталоге:

```
eval zenity --list --text="$word" --width=600 --height=450 \
--checklist --column=# --column=перевод \
"$($echo "$text" | sed "s/^/FALSE /; s/$/ /" | tr '\n' ' ') |
tr '\n' | sed "s/^/$word /" >>~/tetradka
```

«Но только по одной в одни руки!»

А что если мы по ошибке или по забывчивости добавим в нашу тетрадку значение, которое там уже есть? Хорошо было бы автоматически избавляться от таких дубликатов. Чтобы эти дубликаты найти, текст нужно отсортировать. А удалить повторяющиеся строки в отсортированном файле можно либо специальной командой **uniq** (от unique – уникальный), либо соответствующей опцией самой команды **sort**. Называется она в длинном варианте точно так же, как и команда (**uniq**), а в коротком, соответственно, **u**. У этой опции нет гибких возможностей управлять унификацией, как у отдельной команды. Но в нашем случае ее вполне хватит.

По идее, сортировку с унификацией нужно провести в том же самом файле. Но поскольку при перенаправлении ввода и вывода чтение и запись производятся одновременно, то читать и писать один и тот же файл нельзя. Следовательно, нам нужно записать результат во временный файл, а затем этим новым файлом заменить предыдущий. Поскольку в таком варианте имя файла будет упоминаться несколько раз, удобнее будет это имя тоже сохранить в дополнительной переменной. С одной стороны – для краткости и наглядности. А с другой – если мы впоследствии захотим изменить имя или расположение файла, менять его придется только в одном месте: в строке назначения переменной. В скриптах командной оболочки принято такие переменные задавать в начале, опять же для наглядности: чтобы при возможном последующем редактировании знать наперед, где их искать.

```
#!/bin/bash
file=~/tetradka
word="$(xclip -o)"
[...]
eval zenity --list --text="$word" --width=600 --height=450 \
--checklist --column=# --column=перевод \
"$($echo "$text" | sed "s/^/FALSE /; s/$/ /" | tr '\n' ' ') |
tr '\n' | sed "s/^/$word " >>$file
mv $file $file.tmp; sort -u <$file.tmp >$file; rm $file.tmp
```

Раз мы удаляем дубликаты слова с переводом, стоит предусмотреть еще один нюанс: это слово может встречаться в текстах, а значит, и в нашей «тетрадке» – в разном регистре. Например, в начале предложения оно будет написано с заглавной буквы, а в середине – со строчной. Вероятно, возможны и такие случаи, когда все слово будет записано заглавными буквами. Программа **dict** понимает слова вне зависимости от регистра, а вот при сверке дубликатов этот момент будет мешать. Самый простой вариант – автоматически привести все буквы слова к нижнему регистру. В этом нам поможет **sed**: во «второй части» команды **s///** допустим шаблон **\L**, который приводит все следующие за ним символы к нижнему регистру. А следовать за ним должно «все то, что было в первой части команды **s///**». Для этого понятия тоже есть свой шаблон – **&**. Итак, немного модифицируем сохранение слова в начале скрипта:

```
word="$(xclip -o | sed 's/.*\L&/g')"
```

Теперь снова соединяем все вместе (рис. 4).

«Где-то я это уже видел»

Отлично, записывать слова в тетрадку мы уже умеем. Но смысл ее ведь не только в этом. Нужно эти слова оттуда и читать, причем желательно тоже не только вручную. Например, если нам встретится в тексте слово, один или несколько переводов которого уже записаны, то хорошо бы, чтобы наш скрипт сам сказал нам об этом. То есть при получении слова из буфера нужно сначала проверить, есть ли уже такое слово в нашей тетрадке. Если есть, то показать записанные варианты перевода и спросить, записывать ли новый. И только при положительном ответе на вопрос продолжать дальше: отображать полный список переводов с галочками и сохранять выбранные из этого списка.

Чтобы легче было выполнять такую проверку, для начала отделим слово от перевода в тетрадке не пробелом, а неким более уникальным символом, который не будет встречаться в тексте перевода. Для сохранения читабельности тетрадки обычными «ручными» средствами лучше всего подойдет «невидимый» символ – табуляция. Если вы предполагаете и ручное редактирование тетрадки и опасаетесь, что табуляция может быть по недосмотру заменена пробелами, замените ее другим символом или короткой последовательностью символов на свое усмотрение. Для этого придется внести в скрипт чисто косметические правки. А пока вместо пробела вставим после слова табуляцию:

```
eval zenity --list --text="$word" --width=600 --height=450 \
--checklist --column=# --column=перевод \
"$($echo "$text" | sed "s/^/FALSE /; s/$/ /" | tr '\n' ' ') |
tr '\n' | sed "s/^/$word\t/" >>$file
```

Теперь перейдем к вычленению уже записанных переводов и их отображению. Начнем, естественно, с выбора строк с нужным словом:

```
grep "^$word" $file
```

После **\$word** нужно добавить разделитель – не зря ведь мы его вставляли. Тут есть небольшая сложность: команда **grep** не понимает эскейп-последовательности (такие как **\t**, с помощью которой мы обозначали табуляцию). Как из любой безвыходной ситуации, из этой тоже есть как минимум два выхода. Первый – «в лоб»: вписать нужный символ в явном виде. Он не слишком удобен, так как теряется наглядность кода; да и не для любых символов подойдет. Второй способ не настолько прост и очевиден, зато более универсален и читабелен: подставить нужный символ в строку с помощью команды **echo**, которая как раз умеет обрабатывать такие последовательности, если ей задать ключ **-e** (от «escape sequence» – эскейп-последовательность):

```
grep "^$word$(echo -e '\t')" $file
```

Конечно, само переводимое слово отображать в каждой строке незачем, поэтому мы его отрежем командой `cut`:

```
grep "^$word$(echo -e '\t')" $file | cut -f 2-
```

Вас, возможно, удивило, что мы, вопреки предыдущему опыту использования этой команды, на сей раз не задали ей разделитель полей. Ошибки тут нет: именно табуляция выступает разделителем по умолчанию. Сохраним полученное в какой-нибудь переменной; можно в той же самой `text` – она сейчас еще не используется:

```
#!/bin/bash
file=~/.tetradka
word="$(xclip -o | sed 's/^/L&/g')"
```

Далее проверим содержимое этой переменной. И если она не пуста – то есть если строки с таким словом были найдены в тетрадке – то выведем окошко с вопросом, где перечислим найденные варианты. В случае отрицательного ответа на вопрос завершим работу скрипта. А в случае положительного – продолжим, то есть дадим возможность записать еще один вариант. В окошке отобразим: текстовое сообщение; потом отделенный пустой строкой список найденных переводов; а затем, после еще одной пустой строки, вопрос о том, выбирать ли и сохранять ли новый перевод. Поскольку пустые строки мы будем записывать эскейп-последовательностями, которые `zenity`, как и `grep`, тоже не понимает, то воспользуемся еще раз тем же самым приемом:

```
["$text"] && {
zenity --question --text "$(echo -e "Уже записанные переводы слова '$word':\n\n$text\n\nЗаписать новый перевод?")" ||
exit 0
}
```

А вот так (рис. 5) это будет выглядеть на экране.

«А нету ли такого же, но без крыльев?»

Все уже отлично, когда мы можем выделить в тексте словарную форму слова. Но далеко не всегда именно в словарной форме оно там встречается. Как мы видели выше, если в словаре заданное слово не найдено, но найдено несколько похожих, то список этих похожих, предваренный названием словаря, выводится в стандартный поток ошибок. Давайте в этом случае предложим на выбор слова из этого списка и выдадим перевод для выбранного слова. А уж если и похожих не найдется – тогда все-таки выдадим сообщение об ошибке. Делать все это, конечно же, нужно внутри скобок `["$text"] || {..}`. В первую очередь «повернем реки вспять»: поток вывода отправим в никуда, а поток ошибок – на вывод (иначе обработать его будет нельзя):

```
dict "$word" 2>&1 >/dev/null
```

Запись `'&1'` здесь и означает первый поток, то есть стандартный вывод. Важно: перенаправлять потоки нужно именно в такой последовательности. Если сделать наоборот, то к моменту перенаправления ошибок стандартный вывод уже будет «заземлен в пустоту» – и ошибки уйдут туда же.

Далее выбираем нужную строчку по имени словаря и удаляем это имя, оставляя только варианты слова:

```
dict "$word" 2>&1 >/dev/null | grep '^mueller7:' | sed 's/^mueller7: */'
```

И, наконец, приводим список к построчной форме, «сжав» пробелы, а затем заменив их переводами строки. Результат сохраним все в той переменной `text`:

```
text="$(dict "$word" 2>&1 >/dev/null | grep '^mueller7:' |
sed 's/^mueller7: */' | tr -s ' ' | tr '\n' '\n')"
```

Теперь снова проверяем, попало ли что-то в переменную `text`, и на этот раз в случае пустоты уже выдаем ошибку:

```
["$text"] || {
zenity --error --text="Слово '$word' не найдено."
exit 1
}
```

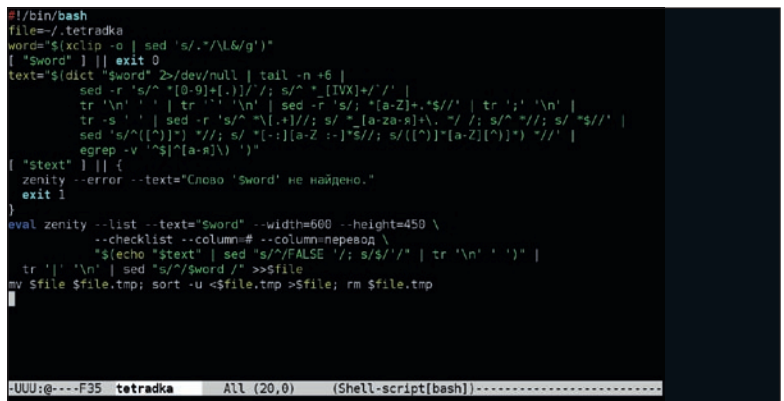


Рис. 4. Промежуточный результат в текстовом редакторе.

Снова почти так же вызываем `zenity` для формирования списка. Только теперь список у нас должен быть не с множественным выбором, а с единственным: не `checklist`, а `radiolist`:

```
eval zenity --list --text="$word" --width=600 --height=450 \
--radiolist --column=# --column=вариант \
"$ (echo "$text" | sed 's/^/FALSE /; s/$/' | tr '\n' ' ')"
```

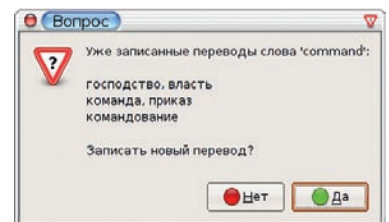
Теперь с полученным словом нужно заново сделать все то же самое, что было описано во всех предыдущих частях статьи. Кто сказал «заново»? Нет и нет! Сейчас мы, подобно барону Мюнхгаузену, вытащим себя за волосы из болота. А именно: скопируем имеющееся слово в тот же иксовый буфер, из которого мы его брали – и вызовем сами себя. «Принцип чайника» в действии. Скопировать можно той же утилитой `xclip` с помощью ее опции `-i` (от input – ввод):

```
eval zenity --list --text="$word" --width=600 --height=450 \
--radiolist --column=# --column=вариант \
"$ (echo "$text" | sed 's/^/FALSE /; s/$/' | tr '\n' ' ')" |
xclip -i
```

А дальше объединим две небольшие хитрости. Помните из первой статьи? Список параметров, переданных функции или скрипту в командной строке, хранится в переменных с цифровыми именами, начиная с `'$1'`. Продолжая эту аналогию, в переменной `'$0'` содержится имя самого скрипта или функции. Таким образом мы можем запустить себя, даже не зная как нас зовут. А запустить себя нам нужно заново – то есть не параллельно с выполняющимся сейчас скриптом, а вместо него. Для этого тоже есть специальная хитрость – встроенная команда оболочки под названием `exec` (от execute – выполнять). Если обычно любая команда, заданная внутри скрипта, запускается «параллельно скрипту» (который ждет ее завершения, а затем продолжает работу), то с `exec` команда будет выполнена *вместо* скрипта. То есть скрипт работу не продолжит, а завершится в этой точке. Этим мы и воспользуемся для финального аккорда:

```
["$text"] || {
text="$(dict "$word" 2>&1 >/dev/null | grep '^mueller7:' | sed
's/^mueller7: */' |
tr -s ' ' | tr '\n' '\n')"
```

Рис. 5. Добавлять ли новый перевод к уже существующим?





Erlang: СУЩНОСТИ

Никто не забыт, и ничто не забыто: завершая рассмотрение базовых сущностей, Андрей Ушаков берется за временно отложенные темы.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

В предыдущих номерах мы поговорили обо всех больших базовых сущностях. На данный момент у нас не рассмотрен ряд тем (из категории базовые сущности), которые либо слишком малы, чтобы про них писать отдельную статью, либо для них не хватило места в одной из статей про базовые сущности (и они оказались менее приоритетными по сравнению с другими темами). Поэтому данная статья посвящена таким темам: атомам, целым и действительным числам, замыканиям и сериализации и десериализации.

Мы уже затрагивали тему атомов в одном из номеров (см. LXF143), теперь вернемся к ней снова. Итак, что такое атомы? В нашем случае, это не те атомы, которые присутствуют в химии и атомной физике; атомы – это аналог констант в языке Erlang. Но это не совсем обычные константы, т.к. обычные константы связывают имя константы с некоторым значением, а в случае атомов имя атома является одновременно и его значением. Так, например, атомами являются величины **true** и **false**.

Какие имена являются действительными именами и, соответственно, значениями атомов в языке Erlang? Если имя начинается со строчной буквы и содержит внутри только буквы (как строчные, так и заглавные), цифры, символ подчеркивания “_” и символ “@”, то такое имя является действительным именем атома. Для всех остальных имен, которые не являются действительными именами атомов, достаточно заключить их в одинарные кавычки, чтобы они стали действительными. Вполне очевидно, что, если имя **name** является действительным именем атома, то и имя **'name'** также является действительным именем атома. Мало того, имена **name** и **'name'** представляют один и тот же атом.

Что важно, атомы в языке Erlang, в отличие от констант, значения которых подставляются компилятором во время компиляции, являются полноправными объектами среды выполнения Erlang. Это означает, что мы можем проверить, является ли объект атомом (**BIF is_atom/1**), а также преобразовывать объекты других типов в атомы и наоборот. Преобразовывать в атомы и наоборот мы можем только строки; мы помним (см. LXF148), что строки у нас могут быть представлены либо в виде списков, либо в виде битовых строк. Поэтому преобразовывать атомы мы можем в списки и битовые строки (**atom_to_binary/2**, **atom_to_list/1**) и наоборот (**binary_to_atom/2**, **list_to_atom/1**). Что интересно, помимо обычных функций преобразования битовых строк и списков в атомы, определены версии этих функций, преобразующие битовые строки и списки в атомы, только если данные атомы уже созданы (**binary_to_existing_atom/2**, **list_to_existing_atom/1**). Если атома, соответствующего строковому представлению не существует, то будет брошена ошибка времени выполнения.

Подобная функциональность нужна для обеспечения безопасности при взаимодействии с внешним миром. Атомы, в отличие от других объектов, создаются навсегда и не уничтожаются сборщиком мусора, если на них нет ссылок. При создании нового атома создается запись в системной таблице атомов. Вполне очевидно, что количество записей в этой таблице ограничено. Поэтому, если неаккуратно десериализовать атомы из строковых данных, приходящих из внешнего мира, то может возникнуть ситуация, когда таблица атомов будет переполнена. После этого в системе нельзя будет создать ни одного атома. Конечно, есть возможность создавать атомы не напрямую, а при десериализации других структур данных, например, кортежей. Если существует такая опасность, то сериализацию и десериализацию объектов при взаимодействии с внешним миром лучше организовать при помощи **BIF term_to_binary/2** и **binary_to_term/2**. Если мы хотим иметь константу в том смысле, в котором она понимается в других языках программирования (т.е. некоторая постоянная величина, связанная с некоторым именем, которое заменяется этой величиной на этапе компиляции), то у нас есть один вариант: использовать директиву препроцессора **define**. Например, мы хотим определить некоторую константу **MAGIC_NUMBER** и затем ее использовать в коде программы; сделать это можно следующим образом:

```
-define(MAGIC_NUMBER, 666999666).
...
SomeFunc(Arg) -> SomeFuncImpl(Arg, ?MAGIC_NUMBER).
```

Видно, что для использования константы (а если быть более точным, то макроса), необходимо использовать специальный синтаксис: перед именем константы должен идти символ “?”. Видно, что привычные для нас константы являются ничем иным, как макроопределениями (как в языке C).

На атомы в языке Erlang возложена еще одна важная ответственность: атомы **true** и **false** представляют соответствующие логические значения (отдельного типа данных для логических значений в языке Erlang нет). Это означает, что везде, где ожидается появление некоторой логической величины, должен в ито-

рассмотрены

ге появиться один из этих атомов (непосредственно либо в результате вычисления выражения). Это касается таких областей, как результат сравнения двух объектов, логические выражения, возвращаемое значения предикатов. Давайте рассмотрим всех их по очереди.

Все объекты в языке Erlang можно сравнивать не только на равенство (оператор `==`) и неравенство (оператор `/=`), но и на упорядоченность друг относительно друга (операторы `<`, `<=`, `>`, `>=`). При этом списки сравниваются поэлементно; у кортежей сначала сравниваются размеры, и если размеры одинаковы, то кортежи сравниваются поэлементно. А что будет, если мы попытаемся сравнить два элемента разного типа? Элементы разных типов считаются не равными, за одним исключением: если сравниваются целые и действительные числа, то целое число преобразуется в действительное, после чего и происходит сравнение. Так, например, выражение `1 == 1.0` вернет `true`.

Если мы будем сравнивать два элемента разных типов на упорядоченность, то все зависит от типов сравниваемых объектов. В языке Erlang, в отличие от многих других языков, сравнение объектов разных типов разрешено и определено. Разные типы данных имеют следующий порядок по возрастанию: числа, атомы, ссылки, порты, идентификаторы процессов `Pid`, кортежи, списки, битовые строки. Раз уж мы упомянули здесь все типы данных в языке Erlang, то следует сказать пару слов о незнакомых нам типах. Ссылка `[reference]` – это тип данных для хранения некоторой уникальной метки; для получения этой метки используется `BIF make_ref/0` (этот `BIF` начинает давать неуникальные значения примерно через 2^{82} вызовов). Порт (или идентификатор порта) – это тип данных для взаимодействия с внешним миром. О портах мы поговорим в будущих статьях. А теперь вернемся к нашим баранам: к упорядочению данных различных типов.

Рассмотрим несколько примеров. Выражение `12 < aa` вернет `true`, т.к. объекты типа число всегда меньше объектов типа атом. Выражение `{aa, 12} < {12, aa, 12}` вернет `true`, т.к. мы сравниваем кортежи разных размеров и кортеж, стоящий слева от оператора, имеет меньший размер; т.е. он меньше кортежа, стоящего справа от оператора. Выражение `{12, aa} < {aa, aa}` вернет `true`, т.к. кортежи имеют равный размер, поэтому мы сравниваем их поэлементно; при этом первый элемент кортежа, стоящего слева от оператора, меньше, чем первый элемент кортежа, стоящего справа от оператора. Выражение `[aa, 2] < [1, 2, 3]` вернет `false`, т.к. списки сравниваются поэлементно независимо от их размера; при этом первый элемент списка, стоящего слева от оператора, больше, чем первый элемент списка, стоящего справа. Выражение `[1, 2] < [1, 2, 3]` вернет `true`, т.к. все элементы списка, расположенного слева от оператора, равны первым двум элементам списка, расположенного справа от оператора, и список, расположенный справа, имеет больший размер.

Язык Erlang имеет еще одну пару операторов для сравнения объектов: это оператор точного равенства `===` и оператор точного неравенства `===/=`. Разница между этими операторами и операторами равенства и неравенства в том, что операторы равенства и неравенства могут преобразовать тип одного из аргументов (целое число в действительное), а операторы точного равенства и неравенства никогда этого не делают. Так, например, выражение `1 === 1.0` равно `true`, а выражение `1 ===/= 1.0` равно `false`.

Пойдем дальше – поговорим о логических выражениях. Логические выражения строятся из других логических выражений, атомов `true` и `false` и выражений, выполнение которых дает один из атомов `true` или `false` (как, например, выражения сравнения). Для построения логических выражений используются следующие операторы: унарное логическое НЕ `not`, логическое И `and`, логическое ИЛИ `or` и логическое ИСКЛЮЧАЮЩЕЕ ИЛИ `xor`. Помимо этих операторов, есть еще операторы «сокращенного» логического И `andalso` и «сокращенного» логического ИЛИ `orelse`. Отличие этих операторов от обычных операторов И и ИЛИ в следующем: операторы `and` и `or` всегда вычисляют оба своих операнда; оператор `andalso` вычисляет второй операнд, только если значение первого операнда равно `true`; оператор `orelse` вычисляет второй операнд, только если значение первого операнда равно `false`. Так, например, предположим, что определена переменная `X` и ее значение равно `0`; тогда логическое выражение `(X /= 0) andalso (1/X > 2)` вернет `false`, а выражение `(X /= 0) and (1/X > 2)` вызовет ошибку времени выполнения (деление на ноль).

И последнее, с чем связаны атомы `true` и `false`: предикаты. Многие библиотечные

«Сравнение объектов разных типов разрешено и определено.»

функции (например, `lists:filter/2`) в качестве одного из своих аргументов ожидают предикат. Предикат – это функция одного аргумента (рассматриваемого в данный момент), которая возвращает либо `true`, либо `false`. Так, например, если мы хотим получить список всех чисел в диапазоне от 1 до 1000, которые делятся на 13 без остатка, то сделать это мы можем следующим способом:

```
lists:filter(fun(Number) -> Number rem 13 == 0 end, lists:seq(1, 1000)).
```

В этом вызове функции `lists:filter/2` первым аргументом стоит анонимная функция-предикат, которая возвращает `true`, если текущее рассматриваемое число делится на 13 без остатка.

А теперь остановим взгляд на целых и действительных числах. Пожалуй, про действительные числа особо и нечего сказать: они занимают в памяти 16 байт (4 слова) в 32-битной версии и 24 байта (3 слова) в 64-битной версии среды выполнения. Для действительных чисел определены стандартные арифметические операторы. И, наконец, в стандартной библиотеке языка Erlang находится модуль `math`, который содержит стандартные математические функции, такие как синус, косинус и т.д.

С целыми числами все намного интереснее. Целые числа делятся на два типа: обычные и большие, причем это деление про-

»

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

исходит «под капотом» (в отличие, например, от языка **Java**, в котором есть обычные типы целых чисел и есть тип **java.math.BigInteger** для больших целых чисел). Мы работаем с большими целыми числами точно так же, как и с обычными. Обычные целые числа занимают 4 байта (одно слово) в 32-битной версии среды выполнения (а если быть точным, то 28 бит); в 64-битной версии среды выполнения они занимают 8 байт (одно слово) – а если быть точным, то 60 бит. Большие целые числа занимают минимум 12 байт в 32-битной версии среды выполнения и минимум 24 байта в 64 битной версии среды выполнения. Сверху большие целые числа ничем не ограничены (точнее, ограничены размером адресного пространства процесса). Такая архитектура целых чисел в языке Erlang позволяет легко решать разные задачи, в которых требуется работа с произвольными целыми числами, например, из области computer science, теории чисел и т.д.

Давайте рассмотрим небольшой пример: предположим, перед нами стоит задача посчитать сумму всех цифр числа **100!** (так, например, **6! = 720**, а сумма цифр числа **6!** равна **9**). Для начала определим функцию вычисления значения факториала (причем определим с использованием хвостовой рекурсии):

```
factorial(0) -> 1;
factorial(1) -> 1;
factorial(N) -> factorial_impl(1, N).
factorial_impl(Value, 1) -> Value;
factorial_impl(Value, N) -> factorial_impl(N*Value, N-1).
```

С помощью функции **factorial/1** мы можем вычислить значение **100!**. Чтобы осознать, насколько это число велико (и убедиться, что с использованием обычных целых чисел эту задачу на других языках программирования так просто не решить), давайте вычислим величину **1.0*factorial(100)**. Получится действительная величина **9,33*10¹⁵⁷**. Теперь вычислим сумму цифр этого числа:

```
lists:sum(lists:map(fun(Char) -> Char-$0 end,
integer_to_list(factorial(100)))).
```

В результате мы получим величину **648**. С той же легкостью мы можем вычислить сумму цифр еще большего числа – **1000!** (это число уже не представимо в виде действительного числа); результатом будет **10539**. Столь же легко решаются и другие задачи с использованием больших целых чисел на языке Erlang.

В одном из недавних номеров (см. **LXF145**) мы говорили о функциях и связанных с ними концепциях функционального программирования. В этом разговоре (в связи с ограничениями на размер статьи) мы отложили в сторону одну важную концепцию – замыкания. У читателей может возникнуть вопрос: если эта концепция важна, то почему она осталась в стороне? Понимание этой концепции и деталей реализации важно, с точки зрения автора, в языках с изменяемыми типами данных, таких как C#, C++, JavaScript, Java. В чисто функциональных языках (и в частности, в Erlang), где изменяемые типы данных отсутствуют, данная концепция тривиальна и не содержит подводных камней.

Итак, что такое замыкание? Замыкание – это «захват» внешнего контекста при создании локального объекта. Под локальным объектом в языке Erlang подразумеваются анонимные функции (в C# это анонимные делегаты и лямбды, в C++ 11 – анонимные функции, в Java – локальные и анонимные классы, в JavaScript – анонимные функции). Внешним контекстом являются локальные переменные и параметры функции, в которой объявляется данный локальный объект (в объектно-ориентированных языках к контексту также относятся поля объекта, которому принадлежит эта внешняя функция).

Рассмотрим небольшой пример, иллюстрирующий эту концепцию. Мы объявляем фабричную функцию, которая конструирует и возвращает анонимную функцию, складывающую два числа:

```
factory(Number) -> fun(X) -> Number + X end.
```

Фабричная функция возвращает анонимную функцию одного аргумента, но эта функция складывает два числа. Откуда же берется второе число? А второе число берется из внешнего по отношению к анонимной функции контекста: вторым числом является аргумент фабричной функции **Number**. Мы можем присвоить построенную функцию переменной **AddFun = factory(10)** и использовать ее: выражение **AddFun(5)** вернет **15**, выражение **AddFun(25)** вернет **35**, и т.д.

Теперь рассмотрим другой пример, когда внешним контекстом является не аргумент внешней функции, а уже объявленная локальная переменная. В этом примере мы объявляем фабричную функцию, которая конструирует и возвращает анонимную функцию для проверки строки на вхождение в нее одной из известных подстрок:

```
factory() ->
  PredefinedStrings = ["abc", "nmo", "xyz"],
  fun(String) ->
    lists:any(fun(PredefinedString) -> string:str(String,
PredefinedString) > 0 end, PredefinedStrings)
end.
```

Список известных подстрок определяется в локальной переменной **PredefinedStrings** фабричной функции и входит во внешний контекст для конструируемой анонимной функции. Теперь (как и в первом примере) мы можем присвоить построенную функцию переменной **StrFun = factory()** и использовать ее: выражение **StrFun("acddc")** вернет **false**, выражение **StrFun("aabcc")** вернет **true**.

С первого взгляда кажется, что все достаточно просто: при создании внутренней функции мы просто копируем значения внешних переменных и параметров функций (если переменная содержит ссылку на объект в куче, то, естественно, копируется ссылка). И для языка Erlang (впрочем, как для любого другого языка с неизменяемым состоянием) это поведение справедливо. Мы могли бы на этом закончить разговор о замыканиях, но давайте пойдем дальше и для большей ясности поговорим о том, какие есть сложности с замыканиями в языках с изменяемым состоянием (таких как C#, C++, JavaScript, Java). И в качестве языка, на котором мы продемонстрируем, как нам все-таки повезло с Erlang, мы возьмем C#. Для всех примеров на C# автор использует Mono как среду времени выполнения и MonoDeveloper как IDE для разработки (кто находится на «темной стороне силы», для примеров на C# может использовать Visual Studio 2008 и выше). Следует сделать еще пару замечаний по поводу компилятора C# в плане замыканий: компилятор C# создает объект некоторого генерируемого класса, который содержит внешний контекст; компилятор C# генерирует класс, который охватывает минимально возможный внешний контекст.

Попробуем на нескольких примерах понять и объяснить получаемое поведение. Начнем с простого примера, в котором мы создаем несколько анонимных делегатов в цикле, после чего поочередно вызываем их:

```
const int count = 2;
System.Action[] actions = new System.Action[count];
for (int i = 0; i < count; ++i)
{
  actions[i] = () => System.Console.WriteLine(i);
}
foreach (System.Action action in actions)
  action();
```

Мы хотим, чтобы каждый созданный делегат вывел на консоль номер итерации цикла, на которой он был создан (т.е. мы ожидаем,

что выведутся числа 0 и 1). Но на самом деле вывод будет совсем другой: а именно, два числа 2 (значение константы `count`).

Сначала получаемое поведение кажется немного странным, но давайте вспомним замечание про поведение компилятора. В данном случае, минимально возможным контекстом является цикл `for`: компилятор генерирует класс, содержащий в качестве поля переменную `i`, по которой происходят итерации цикла, после чего перед циклом создает объект этого сгенерированного класса, и в анонимных делегатах вместо переменной `i` используется поле этого созданного объекта. Поскольку выполнение делегатов происходит уже после цикла, то естественно, что все делегаты получают одно и то же значение – то, которое получило поле `i` сгенерированного объекта, т.е. число 2.

Чтобы получить поведение, которое мы ожидаем, необходимо заставить компилятор минимально возможным внешним контекстом считать тело цикла. Сделать это можно следующим способом: объявить в теле цикла переменную, которой присваивать значение переменной цикла. Это вынудит компилятор сгенерировать класс, который в качестве поля будет содержать эту переменную и создавать объект этого класса (и связывать этот объект с соответствующим анонимным делегатом) для каждой итерации цикла.

```
const int count = 2;
System.Action[] actions = new System.Action[count];
for (int i = 0; i < count; ++i)
{
    int j = i;
    actions[i] = () => System.Console.WriteLine(j);
}
foreach (System.Action action in actions)
    action();
```

В данном варианте поведение будет совпадать с ожидаемым: выведутся числа 0 и 1.

Помимо этого, весьма неочевидного случая, возможна более очевидная ситуация. Предположим, мы создаем некий объект (например, список), затем создаем анонимный делегат, после чего меняем состояние объекта (например, добавляем в список дополнительные элементы). Когда мы вызовем созданный анонимный делегат, то будем иметь дело с ссылкой на измененный объект. Чтобы такого не происходило, нужно создать копию объекта (до создания анонимного делегата) и использовать ее. Пожалуй, это вся «черная магия», связанная с замыканиями в языках с изменяемым состоянием. Но для нас (для тех, кто использует язык Erlang либо другие языки с неизменяемым состоянием) все гораздо проще, и вышеперечисленных подводных камней нет.

А теперь обратимся к сериализации и десериализации данных. Мы уже упоминали о сериализации и десериализации в одном из недавних номеров ([LXF148](#)); давайте вспомним, о чем мы там говорили. Атомы и списки, состоящие из целых чисел, битовых строк и подобных же списков мы можем преобразовывать в обычные битовые строки. Атомы преобразуются в строку; списки же «расплющиваются» и превращаются в битовую строку, которая последовательно содержит все данные в том же порядке, что и в исходном списке.

Вновь рассмотрим несколько примеров. Выражение `atom_to_binary(abc, utf8)` возвращает строковое представление атома в виде битовой строки `<<"abc">>`. Выражение `list_to_binary([1, <<12, 13>>, 2], <<14>>)` «расплющивает» список и возвращает следующую битовую строку `<<1, 12, 13, 2, 14>>`. Битовые строки можно преобразовывать и обратно – и опять же только в атомы и списки. Никакого прямого преобразования в другие типы данных нет. На самом деле отсутствие такой возможности не является чем-то страшным: для работы с низкоуровневыми данными в двоичном формате этих операций, операции создания битовых строк и операции соответствия шаблону [pattern matching] для битовых строк оказывается достаточно (например, для создания клиента, обща-

ющегося через сокеты с внешним сервером). Но предположим, что перед нами стоит задача о сохранении объектов любого типа на внешнем носителе или передача их по сети. Что нам делать в этом случае – разрабатывать свой формат сериализации?

На самом деле ничего подобного делать не надо: в языке Erlang есть сериализация и десериализация объектов любого типа. Для сериализации используется одна из BIF: `term_to_binary/1`, `term_to_binary/2`. Эти BIF сериализуют произвольный объект (при сериализации есть возможность задать уровень сжатия и минимальную версию) и возвращают так называемую расширенную битовую строку. Расширенная битовая строка – это битовая строка, которая помимо самих данных содержит еще и метаданные (и, что важно, длину битовой строки). Так, например, вызов `term_to_binary(1)` вернет следующую битовую строку – `<<131, 97, 1>>`; видно, что битовая строка помимо данных содержит и метаданные. Для десериализации используется одна из BIF: `binary_to_term/1`, `binary_to_term/2`. Эти BIF десериализуют производный объект, позволяя при этом производить безопасную десериализацию. Безопасной называется десериализация, при которой не создаются новые объекты, не собираемые сборщиком мусора. К таким объектам относятся атомы и ссылки на внешние функции. Предположим, например, что в консоли среды выполнения Erlang мы ввели следующее выражение `term_to_binary(ab)`; это выражение будет равно `<<131, 100, 0, 2, 97, 98>>`. Теперь в другой консоли среды выполнения Erlang введем выражение `binary_to_term(<<131,100,0,2,97,98>>, [safe])`. Нашим результатом будет ошибка времени выполнения, т.к. второй экземпляр среды выполнения Erlang ничего не знает про атом `ab`.

В данной статье мы обсудили ряд тем, которые мы еще не рассматривали до этого. Мы прошли большой путь в изучении языка Erlang, но это всего лишь первый шаг. Особая магия языка Erlang заключена все же в области построения многозадачных и распределенных приложений. Тем не менее, без понимания базовых сущностей невозможно понимать и более продвинутые темы. Поэтому следующая статья будет посвящена повторению и закреплению всех тем про базовые сущности; в ней мы сосредоточим свои усилия на практике и на примерах рассмотрим все базовые сущности еще раз. [LXF](#)

Полезные заметки

Директивы препроцессора

- » `-include(File)` Директива для включения содержимого файла `File` в исходный код. Имя файла `File` может быть как абсолютным путем, так и относительным. Имя файла `File` может начинаться с `$VAR`, где `VAR` – это имя переменной среды, при этом `$VAR` заменяется значением этой переменной; если переменная `VAR` не определена, то в имени файла подстрока `$VAR` остается неизменной. Обычно подключаемые файлы содержат определения записей и макросов. Рекомендуется, чтобы имена подключаемых файлов имели расширение `.hrl`.
- » `-include_lib(File)` То же самое, что и директива `include`, только имя файла `File` – всегда путь относительно первого компонента пути, который считается именем приложения (для получения директории для приложения используется вызов `code:lib_dir/1`).
- » `-define(Const, Replacement)` Определение макроса. Используется следующим образом: `?Const`.
- » `-define(Func(Var1, ..., VarN), Replacement)` Определение макроса-функции. Используется следующим образом: `?Func(Var1, ..., VarN)`.

Предопределенные макросы

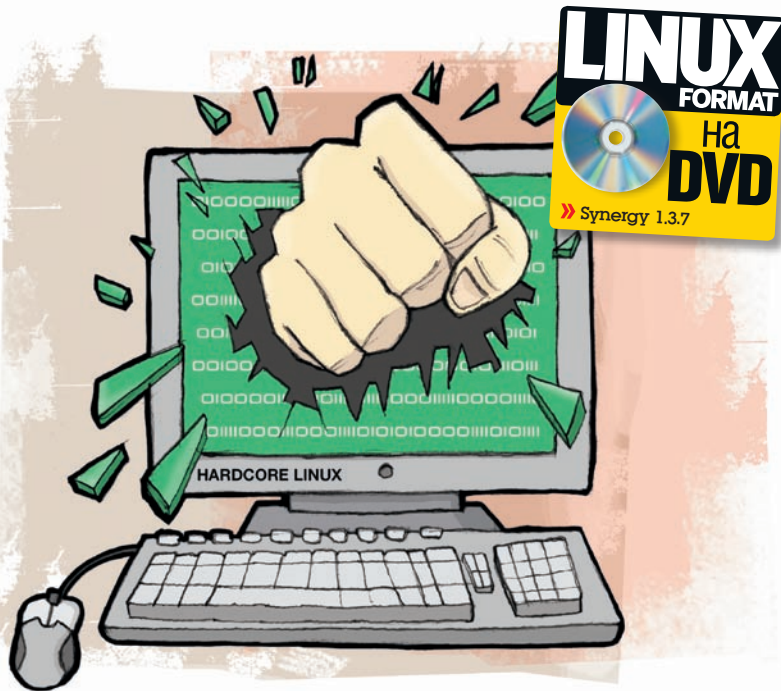
- » `?MODULE` Имя текущего модуля в виде атома
- » `?MODULE_STRING` Имя текущего модуля в виде строки
- » `?FILE` Имя файла текущего модуля
- » `?LINE` Номер текущей строки
- » `?MACHINE` Имя машины, 'BEAM'

Макросы, влияющие на компиляцию

- » `-undef(Macro)` Отмена определения макроса `Macro`.
- » `-ifdef(Macro)` Компиляция последующих строк только в том случае, если макрос `Macro` определен.
- » `-ifndef(Macro)` Компиляция последующих строк, только если макрос `Macro` не определен.
- » `-else` Разрешено применять только после директив `ifdef` или `ifndef`. Если условие (заданное в директиве `ifdef` или `ifndef`) ложное, то компилируются строки после директивы `else`.
- » `-endif` Определяет конец действия директив `ifdef` или `ifndef`.

Synergy: СВЯЗЬ

Алекс Кокс покажет, как за несколько простых шагов увязать все настольные компьютеры – с Linux или нет – в однородную вычислительную среду.



» Три компьютера, одна мышь и одна клавиатура. Переезжать больше не придется...

до Windows и до OS X. Здорово, правда? Можно связать воедино самые разношерстные устройства – ноутбуки, настольные компьютеры, моноблоки, что угодно – и таскать курсор мыши с монитора на монитор. Клавиатура будет работать с тем экраном, где находится курсор мыши.

Существует несколько вариантов первоначальной установки Synergy. Будем считать, что основной компьютер с Linux выступает как сервер, а другие компьютеры являются клиентами, хотя все, о чем мы будем говорить на нашем уроке, столь же справедливо и для случая, когда сервер – это компьютер с OS X или Windows. В работе Synergy с разными платформами отличий очень мало.

В Центре ПО [Software Centre] Ubuntu имеются загрузки для сдвоенного пакета клиент–сервер: просто наберите 'Synergy'

в строке поиска. Это графическая версия, но по сути она (как и пакет **synergy**, который можно установить через **apt-get** из командной строки) лишь устанавливает клиент (**synergyc**) и сервер (**synergys**)

с простым графическим интерфейсом, который запускает их. Нам еще придется создать файлы настройки – тут-то и пойдет потеха.

Затем установите клиент Synergy на каждом из компьютеров, которыми вы намерены пользоваться. Соответствующие установщики для других платформ можно найти на www.synergy-foss.org.

Ваш первый сервер

Начальная настройка Synergy довольно проста. Для нее мы воспользуемся графическим интерфейсом, поэтому сперва откройте графический интерфейс Synergy – если все идет по плану, он будет в списке установленных приложений. Прежде всего убедимся, что все его сетевые настройки пригодны для вашей сети. Активируйте окно и зайдите в Edit > Settings [Редактировать > Настройки]. В открывшемся окне задайте локальный IP-адрес сервера в поле интерфейса и задайте порт, который будет использоваться Synergy.

Одного компьютера человеку мало. Взгляд на офис LXF показывает, что на столах почти у половины сотрудников их по меньшей мере по два, а то и по три. В одних случаях это означает две мыши, две клавиатуры и два монитора. В других – дорогой (и ненадежный) переключатель KVM в действии, разделяющий один набор периферийных устройств между несколькими компьютерами. Но ни одно из этих решений не оптимально.

В первом случае мы имеем перерасход периферии. Зачем держать два набора устройств, если хватает одного? А пара клавиатур неизбежно приведет к ситуации, когда текст, набранный для одного компьютера, окажется на другом. Это неэффективно и неудобно, но по крайней мере оба компьютера доступны одновременно.

Во втором случае переключение между компьютерами неудобное и медленное. Нужно жать кнопку, ждать несколько секунд, снова ждать, пока монитор автоматически определит свои настройки, и еще подождать при переключении обратно. Устройства KVM неудобны и мешают эффективной работе. Наверное, единственный верный способ работы с несколькими компьютерами на одном столе – с одним набором периферийных устройств и несколькими мониторами, и здесь в дело вступает Synergy.

Synergy – клиент-серверная система, которая отправляет сигналы о перемещении мыши и нажатии клавиш клавиатуры с одного сервера на любое число компьютеров сети, и на этих компьютерах могут быть любые операционные системы – от Linux

«Зачем держать два набора устройств, если хватает одного?»



Наш эксперт

Алекс Кокс – зам-редактора журнала PC Plus, страстный коллекционер компьютеров и рьяный сторонник платформенной независимости.

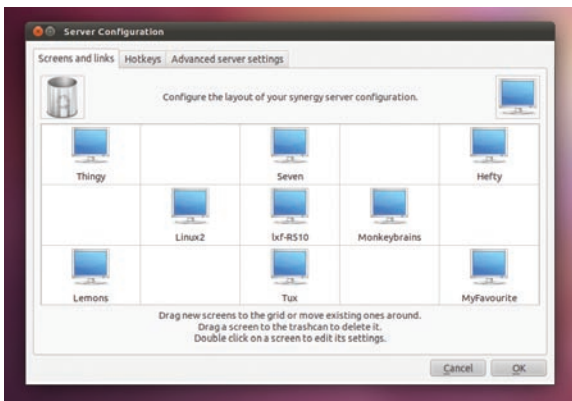
ВАШИХ МАШИН

В этом пункте надо отметить, что *Synergy* – не самая безопасная программа на свете. Выбранный порт нужно открыть в локальной сети, но заблокировать для доступа извне, если вы не хотите, чтобы вашей мышью завладели случайные хакеры. Общего доступа к экрану нет – хотя бы это снижает риск; но нажатия клавиш могут быть считаны, поэтому будьте внимательны при настройке маршрутизатора. Если это вас действительно беспокоит, пропустите *Synergy* через SSH или аналогичный защищенный протокол.

Задайте имя компьютера так, чтобы понять его смысл, когда вы увидите его снова. Мы выбрали **LXF-servermachine** – вы можете выбрать **mainlaptop** или **middle**, в зависимости от постоянства вашей конфигурации. Помните, что настройка через графический интерфейс позволяет быстро добавлять и удалять необходимые компьютеры, но она не настолько гибкая и удобная, как текстовый файл настройки, который мы создадим позже. Нажмите OK и вернитесь в главное окно.

Поставьте галочку рядом с Server [Сервер] в верхней части окна, выделите радиокнопку Configure Interactively [Интерактивная настройка], затем выберите Configure Server [Настроить сервер]. В этом окне вы сможете примерно разместить компьютеры на своем столе в том порядке и в том положении, в котором они на самом деле будут находиться. В центре вы видите имя сервера; перетащите монитор в правом верхнем углу окна в одну из ячеек в центре, чтобы добавить к группе другой компьютер. Задайте его положение по отношению к серверу, затем дважды щелкните на нем и введите имя. Чтобы точно узнать имя, запустите клиент и откройте меню Edit > Settings [Редактировать > Настройки] на втором компьютере – любое указанное здесь имя нужно ввести в настройки сервера. Повторите эти действия для всех ваших компьютеров.

Затем на втором компьютере (у нас это был нетбук с Windows) поставьте галочку рядом с Client [Клиент], а затем воспользуйтесь



» Из нескольких машин можно создавать невероятные комбинации и напрапалуно скакать между ними.

Автоматический запуск

Запуск *Synergy* вручную несколько дискредитирует идею – ведь все должно быть просто. Чтобы запустить сервер *Synergy* при запуске системы, нужно подключить его к *X Window System*, а именно в файл **.xsession**. Найдите файл **.xsession** – обычно он находится в **/etc/X11/** или, если вы пользуетесь KDE, он может быть в **/etc/kde3/** или **/usr/kde/version/share/config** – и откройте его на редактирование с правами суперпользователя. Добавьте в него следующую строку:

```
/usr/bin/synergys <synergy-serverhostname>
```

Делать это нужно в фоновом режиме (или скрипт вообще не завершится, а X-сервер никогда не запустится), поэтому избегайте аргумента **-f**. Эта команда запустит сервер после запуска системы, и вы можете заменить **synergys** на **synergsc**, если хотите запускать клиент автоматически. Если нужно запустить сервер перед входом в систему или если вы хотите запускать ее автоматически на других платформах, зайдите на <http://synergy2.sourceforge.net/autostart.html>. Процедура настройки не очень очевидна, но она должна занять всего пару минут на каждом компьютере.

IP-адресом первого компьютера в качестве имени сервера. Запустите клиент, затем на первом компьютере запустите сервер – вы должны обнаружить, что можете перемещать курсор мыши между мониторами и пользоваться одной клавиатурой на обоих. Легко! Обратите внимание, что для управления клиентским компьютером также можно воспользоваться его собственными мышью и клавиатурой, что удобно для ноутбуков.

Разворачиваем свой

Итак, это работает. Если все, чего вы хотели – перемещать мышью между мониторами, можете здесь и остановиться. Но для *Synergy* это далеко не предел, если уметь им манипулировать. Большая проблема с графической утилитой настройки на данный момент – недостаток контроля. Гораздо больше настроек можно задать в текстовом файле настройки – и даже пересылать его между компьютерами, чтобы в вашем гареме любой мог выступать в роли сервера – но нужно создать этот файл с нуля самим.

Откройте любимый текстовый редактор и приступите к созданию файла – как вы его назовете и где сохраните, неважно: это простой текстовый файл. Прежде всего опишите связанные компьютеры сети в следующем формате:

```
section: screens
linux:
windows:
mac:
end
```

Каждая часть файла настройки начинается с заголовка **section:** и заканчивается словом **end**. Здесь можно использовать любые имена – главное, чтобы вы могли по имени распознать компьютер. Настоящие имена хостов мы добавим позже.

Скорая помощь

Не видите графического интерфейса? Это новое добавление к 1.4 beta, а на момент написания статьи стабильная версия *Synergy* – лишь 1.3. Если вам нравится перетаскивание мышью, зайдите на сайт *Synergy* и установите бета-версию.

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



Скорая помощь

Текущая версия Synergy не очень хорошо работает с графическими планшетами. Если вы хотите пользоваться любимым устройством Wacom, вставьте его в клиентский компьютер и примените его собственные драйверы, чтобы со всех сторон окружить используемую область экрана рамкой в один пиксель, во избежание случайного переключения экрана.

Такое задание мониторов, с двоеточиями, позволяет указать дополнительные аргументы. Например, можно переключить клавиши Ctrl и Alt на Mac (чтобы горячие клавиши были одинаковы во всех платформах), добавив `ctrl = alt` и `alt = ctrl` отдельными строками после обозначения `mac`:

Также можно снабдить эти компьютеры полудуплексными клавишами блокировки, т.е. клавишами, которые при нажатии не отправляют события «нажато» и «отпущено», а вместо этого отправляют одно событие «нажато» при включении и одно событие «отпущено» при выключении. Для этого установите параметры `halfDuplexCapsLock`, `halfDuplexScrollLock` и `halfDuplexNumLock` в `true` или `false`. Помните, что в файле настройки регистр имеет значение, поэтому вводите все так, как здесь написано.

Назначение следующего раздела файла настройки довольно очевидно – в нем пора задать настоящие имена компьютеров, которым мы на время дали простые имена. Скопируйте и вставьте в него структуры из раздела `screens`, выкиньте все, кроме имен машин, и переименуйте раздел в `aliases` [псевдонимы].

При каждом имени компьютера укажите его имя хоста – чтобы его узнать, наберите `uname` в окне терминала. Если у вас сеть с доменом, можно также включить сюда доменный адрес компьютера или воспользоваться вместо него IP-адресом. Последний вариант менее надежен, так как IP-адреса имеют свойство изменяться.

Добавляя Mac, помните, что к его имени хоста часто нужно приписывать `.local`, чтобы оно верно распознавалось. Если оно не работает, причина именно в этом.

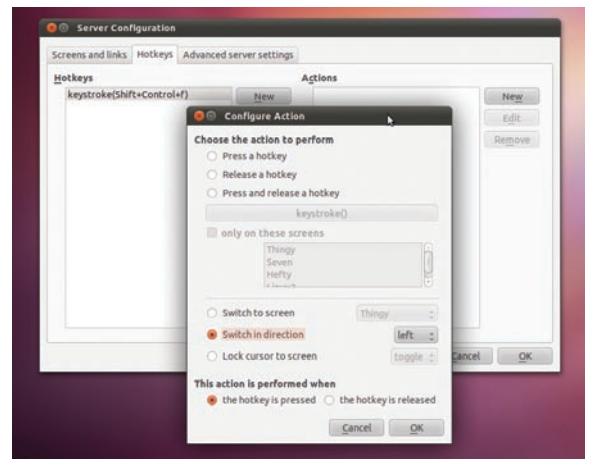
Связи

Вот интересный момент и настоящая причина воспользоваться текстовым файлом настройки: связи. Всего несколькими строками можно определить точную структуру сети компьютеров, учитывая высоту экрана до физического расположения.

Опять скопируйте раздел `screens`, как вы проделали с разделом `aliases`, и назовите его `links`. Теперь укажите информацию о взаимном расположении мониторов относительно друг друга. Если компьютер с Windows стоит справа от компьютера с Linux, добавьте строку `right = windows` после заголовка `linux`. Это определяет взаимодействие между компьютерами в одном направлении – переместите курсор за правую границу монитора компьютера с Linux, и он появится на рабочем столе Windows, но в обратном направлении это не будет работать, пока вы не включите противоположную строку после заголовка `windows`:

Обратите внимание и на такой случай:

```
section: links
linux:
right(0,50) = windows
```



➤ Комбинации клавиш можно задать и через GUI, но мы сочли, что проще добавить их в файл настройки.

```
windows:
right(50,100) = mac(0,50)
mac:
right = linux
end
```

Этот код по сути создает цикл – продолжайте перемещать курсор вправо (с Linux на Windows и на Mac), и вы снова окажетесь на компьютере с Linux – но при движении влево это работать не будет. Если хотите, можете также оставить один компьютер без связей. Тогда курсор застрянет на его мониторе; но, как мы узнаем позже, есть способы это обойти.

Если вы не богаты или у вас нет знакомых на заводе-изготовителе мониторов, все ваши мониторы будут разных размеров и разной высоты. Обратите внимание на числа в скобках в приведенном выше коде: они преобразуются в пару процентных величин по отношению к высоте экрана, начиная сверху или слева и заканчивая снизу или справа в зависимости от края монитора. В приведенном примере монитор Windows расположен рядом с правой верхней половиной монитора Linux, а левая верхняя половина монитора Mac расположена рядом с правой нижней половиной монитора Windows. С помощью этих параметров не нужно задавать реальное физическое расположение (например, попробуйте что-нибудь вроде `(95,100)`, чтобы задать «горячий угол»), и можно даже задать несколько путей, ведущих от одной стороны экрана, путем разбивания их путей.

К сожалению, мы ничего не можем сказать о том, как задавать ваши собственные проценты. Над этим придется покопаться – и покопаться на каждой стороне – прежде чем мышь начнет плавно снова по мониторам. Впрочем, результат стоит усилий.

Личное поведение

Перемещение между мониторами – это хорошо, но Synergy способен на большее. В его опциях можно задать точнейшие настройки. Для начала – ну да, создадим новый раздел, и назовем его, да, `options`. Имя компьютера на сей раз не потребуется – эти настройки являются глобальными; просто укажите одну или несколько из перечисленных далее опций со знаками равенства и любыми необходимыми значениями, каждый параметр с новой строки.

С помощью параметра `heartbeat`, за которым идет число, можно определить интервал времени, через который клиент должен отправлять обновление статуса на сервер, в миллисекундах; если сервер не получит обновления статуса три раза, клиент будет отключен.

Защищенное соединение

Хотите быстро и легко запустить защищенное соединение? Можете потратить несколько часов на настройку SSH на работу с Synergy – и кое-как заставить SSH работать в Windows, если придется переключаться между различными ОС – или воспользуйтесь Stunnel, кросс-платформенной программой, которая прекрасно это делает.

Установите Stunnel и задайте следующую конфигурацию на сервере:

```
output = stunnel.log
socket = !:TCP_NODELAY=1
socket = r:TCP_NODELAY=1
```

```
cert = stunnel.pem
[synergy]
accept = 25800
connect = 24800
```

Затем сделайте то же самое на сервере, но измените `accept` на `24800`, добавьте в начало файла строку `client = yes` и установите `connect` в `<server hostname>:25800`. Теперь просто запустите `stunnel` и `synergys` на сервере, `stunnel` на клиенте и укажите `synergys` на localhost (который перехватывает порт Synergy), а не на имя хоста. Теперь вы должны работать по защищенному соединению.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Чтобы задать «холодные углы», при попадании курсора мыши в которые монитор не будет сменяться, воспользуйтесь параметром `switchCorners` с набором наименований углов (`top-left`, `bottom-right` и т. д.). Здесь также можно указать знаки плюс и минус; чтобы сделать активными все углы кроме верхних и правых нижних, попробуйте `switchCorners = all -right`. Размер «горячего угла» в пикселях задает параметр `switchCornerSize = x`. Помните, что эти параметры применимы и к отдельным машинам – просто переместите команду в раздел `screens` соответствующей машины.

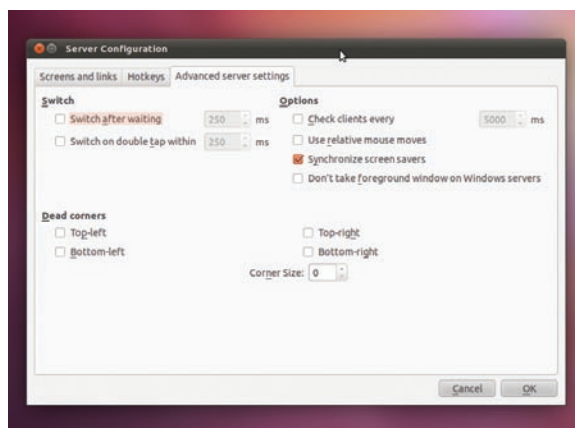
С помощью параметра `switchDelay` можно задать число миллисекунд, в течение которых курсор нужно держать у края одного монитора, прежде чем он переместится на следующий. Это позволит избежать случайных переключений. Также можно воспользоваться командой `switchDoubleTap` (опять же с интервалом времени в миллисекундах) чтобы для переключения требовалось двойное касание курсором края монитора в течение заданного интервала времени. Можно использовать оба параметра одновременно.

Вот интересный параметр; хотя с помощью *Synergy* нельзя разделять экраны, она неплохо справляется с тем, чтобы все компьютеры воспринимались как один. Как насчет синхронизации ваших хранителей экрана с тем, что выполняется на сервере? Для этого просто добавьте параметр `screenSaverSync = true`. Помните, что нажатие любой клавиши – даже на клиентских компьютерах – разорвет синхронизацию.

Нажмите клавишу

Последнее и самое нужное для тех, кто предпочитает клавиатуру мыши – комбинации клавиш. По умолчанию для привязки мыши и клавиатуры к одному монитору в *Synergy* используется тоскующая без работы клавиша `Scroll Lock`. Независимо от раскладки клавиатуры нажмите `Scroll Lock` – и периферийные устройства останутся на компьютере, которым вы сейчас пользуетесь.

В разделе `options` также можно определить особые клавиатурные комбинации и даже изменить те, что уже там есть. Например, если вы хотите изменить клавиши, привязывающие периферийные устройства к одному монитору, сперва отключите функции по умолчанию (`keystroke(ScrollLock) = lockCursorToScreen(off)`), а затем задайте другую клавишу или комбинацию клавиш командой (`Ctrl+Shift+F1`) = `lockCursorToScreen(toggle)`. Также можно использовать `on` вместо `toggle`; в этом случае нужно задать еще одну комбинацию клавиш для выключения. Функцию `keystroke` можно использовать для отправки нажатия или комбинации клавиш всем компьютерам одновременно. Взгляните на следующий пример:



➤ Параметров настройки сервера бездна, поэтому готовьтесь к многочисленным прогам и ошибкам.

Немного истории

Synergy появилась в 1996 году, когда несколько разработчиков не существующей ныне *Cosmo Software* создали *CosmoSynergy*, хитрый способ одновременного использования компьютеров с *Windows* и *Irix* без дорогого и неуклюжего *KVM*. *CosmoSynergy* так и не был должным образом выпущен; *Synergy* – полная переработка с нуля.

С тех пор программа пережила несколько разветвлений. Начальная версия, выпущенная в 2001 году, поддерживала только *X Window System*, а поддержка других систем была запланирована на более позднее время. Разработка остановилась в 2006 году – хотя проект никогда официально не закрывался – и рабочее ответвление, *Synergy+*, стартовало в 2009 году. Позже, в 2010 году, команда

Synergy+ попыталась официально взять проект на себя, но его создатель Крис Шенеман (*Chris Schoeneman*) отказался, предпочтя объединить два проекта официально и создать то, что ныне известно под названием *Synergyfoss* – см. сайт <http://synergy-foss.org>.

Что же дальше? Определенно, чтобы избежать мелких ошибок в текущей версии, нужно полностью переписать программу. Выход *Synergy 2.0* запланирован на начало 2012 года и построен вокруг методологии *Scrum*, но разработчики обещают сохранить совместимость с предыдущими версиями по файлам настройки. Также будут предусмотрены встроенное шифрование, поддержка копирования файлов между компьютерами и другие функции.

```
keystroke(L) = keystroke(ctrl+alt+delete,windows),
keystroke(super+F1,mac)
```

Эта непрактичная и бесполезная команда в одну строку преобразует любое нажатие клавиши `L` в `Ctrl Alt Delete` на компьютере *Windows* и в то же самое время в `Super+F1` на компьютере *Mac*. Также можно разбивать команды точкой с запятой вместо запятой, чтобы обозначить, что происходит, когда клавиша нажимается и отпускается:

```
keystroke(shift+A) = mousebutton(1) ; mousebutton(2)
```

Эта команда отправит сигнал нажатия левой кнопки мыши при первоначальном нажатии `Shift+A` и средней кнопки мыши при отпускании этих клавиш. Также можно указать точку с запятой, но опустить один из аргументов, чтобы отдельно задать действия, происходящие при нажатии или при отпускании клавиши.

Наконец, можно самому задать команды переключения мониторов и либо переключаться на конкретный компьютер, либо выбирать компьютер по его расположению. Просто воспользуйтесь модификаторами `switchToScreen(screenname)` для перехода на конкретный компьютер или `switchInDirection(direction)` для перехода по заданному пути. Если вы планируете использовать этот метод, вероятно, стоит избегать наличия нескольких связей в одном направлении, так как это может сбить с толку *Synergy*.

Закруглимся

На данный момент у вас должен быть файл настройки с четырьмя различными разделами, заполненными всевозможными командами. Резвиться с настройками можно бесконечно, но теперь пора сохранить файл: сохраните его в `/etc/` под именем `synergy.conf`. Теперь можно запустить сервер через графический интерфейс (убедитесь, что выбрали правильный путь в соответствующем окне) или с командной строки.

В последнем случае, если файл сохранен в правильном месте с верным именем, достаточно просто выполнить команду `synergys`. Можно запустить сервер на переднем плане, командой `synergys -f`. На других компьютерах запустите `synergyc` с именем хоста сервера в качестве аргумента или введите имя хоста (или имя домена или IP-адрес в зависимости от конфигурации сети) в соответствующем окне. Если все прошло нормально, ваш сервер *Synergy* должен запуститься. Теперь вас ограничивает только собственное воображение. **LXF**

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru

В этом месяце мы ответим на вопросы про...

- 1) VT HomeHub2
- 2) Шалости KDE
- 3) Перемещение музыкальных файлов
- 4) Прошивки для беспроводных устройств
- 5) Plymouth
- 6) Перемещение почты
- 7) Сканеры пленки и VueScan
- 8) Сломанный OpenSUSE

1 Подключение HomeHub

В недавно я установил на свой Dell Optiplex GX620 Linux Ubuntu 10.10 вместе с Windows XP. Несколько раз попытался выйти в Интернет, но позорно провалился! Я пользуюсь VT HomeHub2, подключенным к компьютеру через Ethernet.

Я «пенсиянер», и старые серые клеточки работают не слишком хорошо. Не могли бы вы мне помочь? Простым — не очень техническим — языком.
Питер Уайт (Peter)

Ваш HomeHub — модем и маршрутизатор, который выступает в качестве интерфейса между двумя сетями: локальной сетью (LAN) и сетью Вашего провайдера. Прежде всего убедитесь, что локальная сеть ра-

ботает — даже если компьютер всего один, сеть все равно есть, так как маршрутизатор тоже является сетевым устройством. У маршрутизатора есть IP-адрес, и, судя по предоставленной Вами информации, это 192.168.1.254; поэтому первый шаг — попробовать подключиться к нему, открыв в браузере <http://192.168.1.254>. Должна открыться страница администрирования маршрутизатора (сначала, вероятно, вас спросят пароль администратора). То же самое должно происходить при подключении к маршрутизатору из Windows.

Если подключиться не удастся, убедитесь, что Вы используете верный адрес — он должен быть указан в документации, и его также можно посмотреть в Windows — и попробуйте связаться с ним, открыв терминал и выполнив команду:

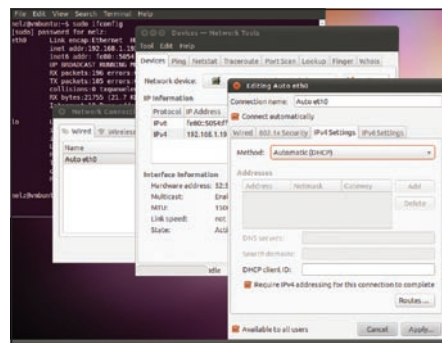
```
ping -c 3 192.168.1.254
```

Если это работает и Вы точно знаете, что маршрутизатор подключен к Интернету, потому что он работает в Windows, проверить нужно несколько вещей. Первая — убедиться, что сетевой интерфейс использует верный маршрут к Интернету. Откройте терминал и наберите

```
sudo route -n
```

Последняя строка должна быть примерно такой:

```
0.0.0.0 192.168.1.254 0.0.0.0 UG 2 0 0 eth0
```



Если Ubuntu не подключается к Интернету по проводному соединению, ваша первая остановка — **Network Tools**.

0.0.0.0 означает любой адрес, а следующий адрес должен быть адресом Вашего маршрутизатора. Если это работает, наберите

```
ping -c 3 www.google.com
```

Команда должна вывести некоторую статистику, и самая важная для Вас строка —

```
3 packets transmitted, 3 received, 0% packet loss
```

Она гласит, что все входящие запросы получили ответ. Если это не сработает, замените www.google.com на 209.85.227.104, IP-адрес для www.google.com. Если это работает, а имя — нет, у Вас проблема с настройками DNS. Они задаются в файле `/etc/resolv.conf`, поэтому вернитесь в терминал и скопируйте

```
cat /etc/resolv.conf
```

Здесь должен быть указан сервер имен для адреса маршрутизатора (маршрутизатор также выступает в качестве сервера имен Вашей сети). Если его нет или ничего по-прежнему не работает, перезагрузитесь в Windows, откройте сетевые настройки и запишите IP-адрес, шлюз и DNS-адреса. Затем вернитесь в Ubuntu, запустите **Network Tools** из меню System > Administration [Система > Администрирование], выберите интерфейс Ethernet, нажмите **Configure [Настройка]**, затем **Edit [Изменить]**, перейдите на вкладку IPv4 и переключитесь с Automatic [Автоматически] на Manual [Вручную]. Нажмите Add [Добавить] и введите записанные IP-адрес и шлюз, затем добавьте DNS-сервер в окошке ниже. Перейдите на вкладку IPv6 и убедитесь, что она установлена в Ignore [Игнорировать].

Если к маршрутизатору в браузере не подключиться, у Вас серьезные ошибки в настройках сетевого интерфейса. Запустите **Network Tools** из меню System > Administration [Система > Администрирование], удалите существующее интернет-соединение и создайте новое, с автоматическими настройками. **НБ**

Наши эксперты

Мы подыщем эксперта на любой ваш вопрос. От установки и проблем с модемом до администрирования сети — ответ найдется! Просто напишите нам бумажное или электронное письмо, и о нем позаботятся.



Нейл Ботвик

Владелец ISP и экс-редактор диска для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс

Майк был одним из создателей прототипа LXF — Linux Answers. Его специальности — программирование, оконные менеджеры, скрипты инициализации и SNES.



Джонатан Робертс

Джонатан столько отлынивал от занятий ради установки, настройки и взлома серверов, что теперь он эксперт.



Грэм Моррисон

Когда Грэм не обозревает кучи программного обеспечения и не халтурит с *MythTV*, он готов к ответам насчет любого оборудования и проблем виртуализации.



Валентин Синецын

В редкие свободные минуты Валентин обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема — настольный Linux.



Юлия Дронова

Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования www.unixforum.org.

Куда посылать вопросы

Пишите нам по адресу answers@linuxformat.ru или спрашивайте на форуме: www.linuxformat.ru

2 Сброс KDE

В Недавно я перешел с Ubuntu 10.10 на Kubuntu 11.04 на своем основном компьютере, и мне понравилось это изменение. Я решил проделать то же на своем ноутбуке. Очень давно (несколько месяцев, а не лет тому назад) я установил KDE и решил попробовать его, прежде чем удалять Gnome. Нижняя панель была сверху, главное меню заполняло экран и было весьма хаотичским. Не знаю, с чего это KDE стал таким.

Я хочу сбросить KDE в настройки по умолчанию, но подозреваю, что при повторной установке все будет выглядеть точно так же (у меня отдельный раздел для /home). Я не в восторге от полной переустановки, потому что у меня на жестком диске много всего. Я бы лучше сбросил KDE, чтобы можно было его обновить, или установил бы Kubuntu 11.04 поверх раздела / и сохранил бы /home. Можно ли сбросить KDE без очистки /home?

heioiwge

Переустановка KDE ничего не изменит, так как его внешний вид определяется настройками в каталоге **home**. Они находятся либо в `~/.kde`, либо в `~/.kde4`, в зависимости от Вашего дистрибутива. Удалите или переименуйте этот каталог, и при следующем входе в систему Вы вернетесь к настройкам по умолчанию. Это удалит все Ваши настройки KDE, включая всю почту в *KMail* и все сохраненные пароли в *KWallet*. Если Вы переименуете, а не удалите каталог, то после настройки внешнего вида рабочего стола сможете скопировать необходимые файлы обратно.

Альтернатива – удалить только файлы, отвечающие за внешний вид рабочего стола. Конфигурационных файлов в `~/.kde` много, и есть риск пропустить один-два, но важно удалить файлы `~/.kde/share/config/plasma-desktoprc` и `~/.kde/share/config/plasma-desktop-appletsrc`.

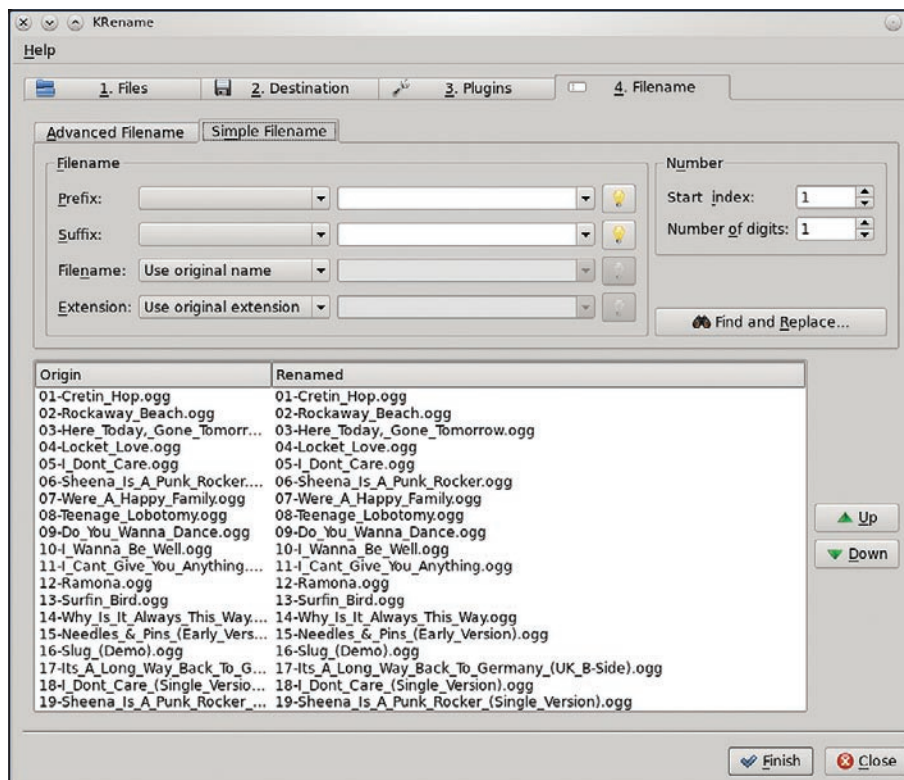
Лучше делать это, когда Вы не работаете в KDE, в противном случае неверная конфигурация все еще будет находиться в памяти, и некоторая ее часть может записаться на диск при выходе из системы. Для этого либо запустите команду `sudo /etc/init.d/kdm stop` в терминале, зайдя в систему от своего имени, либо зайдите в систему как другой пользователь. В последнем случае Вам нужен пользователь с правами на *sudo*, чтобы удалять файлы из домашнего каталога другого пользователя. **MC**

3 Просто как abcde

В Я пользовался *Abcde* для конвертации коллекций своих записей в **FLAC** и **Ogg**. Проблема в том, что *Abcde* помещает все файлы в один и тот же каталог, а я хочу разнести версии **FLAC** и **Ogg** по разным каталогам.

Моя музыка размещена таким образом: `flac/<Artist>/<Album>/xy.<track_name>.[ogg|flac]`. И мне нужно переместить файлы **Ogg** в каталог **ogg** и сохранить структуру каталогов с исполнителями и альбомами.

Я обнаружил, что это можно сделать командой `cp --parent`, но я хочу сделать это со всеми файлами сразу. Я подумал, что мне поможет `find . - name`



» *KRename* упрощает пакетное переименование файлов, но можно настроить свою программу так, чтобы они именовались правильно с самого начала.

“*.ogg” | xargs cp ... но как получить вход канала, чтобы воспользоваться им как источником для команды cp? Это возможно? Какое-то другое решение?

shaddack

Есть несколько способов переместить файлы так, как Вам нужно, но без них можно обойтись, так как в *Abcde* можно с самого начала настроить необходимое размещение файлов. Три важных параметра в `~/.abcde.conf` – **OUTPUTDIR**, **OUTPUTTYPE** и **OUTPUTFORMAT**. Первый – корневой каталог для всей музыки, скажем, `~/music`. Второй – список форматов создаваемых файлов, в данном случае **flac** и **ogg**, а третий – формат именования файла и каталога для хранения каждого файла в **OUTPUTDIR**.

Он состоит из нескольких переменных, задаваемых для каждого трека, таких как **\$ARTISTFILE** и **\$ALBUMFILE**. Для нас важна переменная **\$OUTPUT**, которая задает тип текущего файла – скажем, **flac** или **ogg**.

```
OUTPUTDIR=~/.music"
OUTPUTTYPE="flac,ogg"
OUTPUTFORMAT='${OUTPUT}/${ARTISTFILE}/${ALBUMFILE}/${TRACKNUM}-${TRACKFILE}'
```

С этими настройками файлы **flac** будут создаваться в `~/music/flac`, а файлы **ogg** – в `~/music/ogg`.

Конечно, есть другие ситуации, когда нужно переименовывать файлы, а с этими настройками имеющиеся файлы перемещаться не будут. Существуют графические утилиты пакетного переименования файлов, такие как *KRename* и *Gnome Commander* – в зависимости от дистрибутива. Сделать это можно и из командной строки. Соз-

дать список файлов для переименования можно командой *find*, но в *Bash* теперь есть шаблон ******, который не так давно появился в *Zsh*. ****** соответствует полным путям, а не именам одного файла или каталога в виде *****. Для выбора всех файлов **Ogg** в `~/music` Вам потребуется всего лишь шаблон

```
~/music/**/*.*ogg
```

Хотя ****** по умолчанию активирован в *Zsh*, он может работать в *Bash*; в этом случае его нужно включить командой:

```
shopt -s globstar
```

Это позволяет отказаться от каналов, так как мы и без них сможем пройти по всем файлам. Например:

```
for FILE in **/*.ogg
do
  mkdir -p "ogg/${dirname \"$FILE\"}"
  mv "$FILE" "ogg/$FILE"
done
```

Обилие кавычек и экранированных кавычек необходимо для работы с именами файлов или каталогов, содержащими пробелы или другие спецсимволы. Если в именах Ваших файлов нет пробелов, то, опустив кавычки, Вы получите более читаемый, хотя и менее универсальный, альтернативный вариант.

```
for FILE in **/*.ogg
do
  mkdir -p ogg/${dirname $FILE}
  mv $FILE ogg/$FILE
done
```

Так как **mv** не создает необходимые ей каталоги, сначала нужно вызвать **mkdir**. Команда **dirname** возвращает полный путь до **\$FILE** без имени файла. **FM**

4 Прошивка для беспроводной сети

Мне нравится Debian, но я почему-то не могу настроить беспроводную сеть на ThinkPad T42. Прилагаю вывод *HardInfo*. Я тыкал пальцем в небо, перепробовав все средства, вроде добавления сетевого менеджера *Wicd*, чтобы он выполнял обычные задачи сетевого менеджера. Я добавил модуль микросхемы/драйвер Intel IPW2100 и несколько раз входил и выходил. Выдаваемая ошибка — неверный пароль, но я проверял его миллион раз, и с менеджером по умолчанию у меня была другая ошибка. Я точно не знаю, как удалить сетевой менеджер *Wicd*. К тому же он мне нравится.

Дерри Мэнли [Derry Manley]

Удаление *Wicd* Вас не спасет: проблема находится на более низком уровне, и когда все заработает, Вы останетесь с тем сетевым менеджером, который Вам нравится. Драйверы для микросхем IPW2100 и IPW2200 входят в состав ядра Linux, но для их работы требуется несвободный (как слово) пакет прошивки. Прежде чем ею заняться, проверьте самое простое: включена ли беспроводная сетевая карта? Нажатие Fn-F7 включает и выключает ее.

Нужная Вам прошивка находится в пакете `firmware-ipw2x00` (годном для чипсетов 2100 и 2200) из несвободного репозитория, который нужно предварительно подключить. Добавьте такую строку в `/etc/apt/sources.list`:

```
deb http://ftp.uk.debian.org/debian squeeze main contrib non-free
```

изменив `uk` на локальное зеркало Debian, или поставьте галочки для несвободных репозиторий в *Synaptic*. Перезагрузите репозитории (или запустите `apt-get update`), и `firmware-ipw2x00` появится в поиске. После установки перезагрузите модуль `ipw2100`, либо выгрузив и вновь загрузив его командами

```
rmmod ipw2100
modprobe ipw2100
```

либо перезагрузив систему; это должно инициировать загрузку прошивки и активацию карты. Затем она появится в *Wicd*, и Вы сможете настроить ее как надо. Если это Ваша собственная сеть и тем не менее идут ошибки с паролями, попробуйте отключить шифрование и установить соединение. Так Вы убедитесь, что карта работает правильно до включения и настройки шифрования. В любом случае, не оставляйте свою сеть незащищенной более чем на несколько минут, необходимых для проверки системы. ДжР

5 Насильное удаление

Я работаю в Ubuntu 11.04 со множеством установленных программ — *Twm*, *Fluxbox*, игры, *Qt* и т.д. По какой-то причине *Plymouth* (программа, создающая красивую загрузочную анимацию) не работает, и при запуске компьютера я вижу только текст “Ubuntu 11.04” в правом верхнем углу экрана, но когда я выключаю компьютер — все работает! Мой монитор — Asus, 1366×768.

Теперь я хотел бы (полностью) переустановить *Plymouth* командами `apt-get remove --purge plymouth` и `apt-get install plymouth`, но буквально все в Ubuntu почему-то зависит от него.

Итак, как мне удалить весь этот мусор и решить проблему с *Plymouth*?

Саймон Шилдз [Simon Shields]

Не уверен, что авторам *Fluxbox*, *Qt* и прочего понравится, что Вы назвали плоды их трудов «мусором», но могу понять, что Вам все это не нужно.

Одна из причин простоты и быстрейшего действия установщика Ubuntu в том, что он не досаждал вопросам о том, что Вы хотите установить, а просто копирует все из файловой системы Live CD на жесткий диск.

Недостаток данного подхода в том, что в результате у Вас оказываются лишние программы, и для избавления от них нужно приложить некоторые усилия.

Однако на стандартном CD 11.04 нет *Fluxbox* и *Twm*, хотя есть *Qt*, поскольку она нужна некоторым программам. Наверное, Вы устанавливали систему с нестандартного диска, вроде специальных дисков *LXF* от Майка?

Plymouth нужен такому количеству программ, потому что от него зависит *Ubuntu-desktop*, и все привязано к нему. Его нельзя удалить через `apt-get`, но `dpkg` предоставляет больший контроль (и больше шансов сломать систему). Можно выполнить принудительное удаление командами `sudo dpkg --force-depends --purge libplymouth2 plymouth plymouth-label plymouth-theme-Ubuntu-logo plymouththeme-Ubuntu-text` чтобы удалить все связанные с *Plymouth* пакеты.

Потом выполните команду

```
apt-get --fix-broken upgrade
```

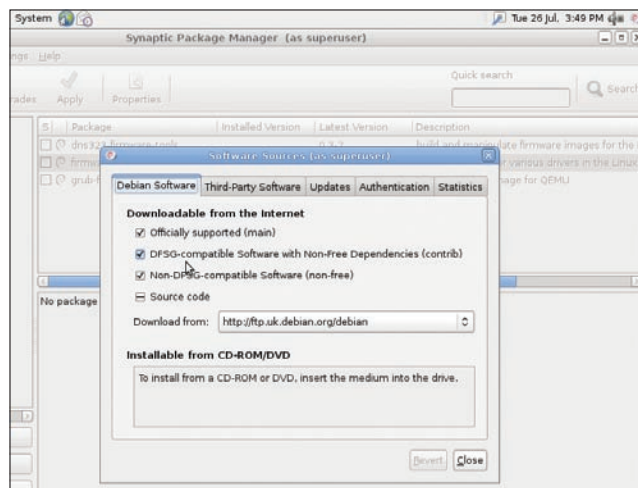
Перед запуском каждой из этих команд к ним можно добавить `--dry-run`. Список всех установленных пакетов с *Plymouth* в имени Вы получите командой

```
dpkg-query --show --showformat='${Package}\n' '*plymouth*'
```

Важно указать кавычки, и они должны быть одиночными, чтобы оболочка не пыталась интерпретировать символы `$` и `*`.

Однако это может и не решить проблемы. Загрузочный экран можно полностью отключить, удалив слово `splash` в строке загрузки в меню загрузчика. Проблему также может вызывать выбранный режим ядра для Вашего монитора — тогда нужно лишь добавить `nomodeset` в параметры загрузки.

Для удаления лишних пакетов пройдите по списку установленных пакетов в *Synaptic*, пометьте для удаления ненужные Вам, а затем по-



Файлы прошивок — только двоичные, и для их установки нужно включить несвободные репозитории Debian.

смотрите, что удалится вместе с ними. Если ничего из этого Вам не нужно, завершите удаление. Проще, но дольше сделать это с несколькими пакетами сразу, но сперва спросите себя: стоит ли игра свеч? Ненужные пакеты занимают место на диске, но не тратят других ресурсов, если их не запускать. Если у Вас нет реальной нехватки дисковой памяти, как, например, в твердотельном накопителе, спокойнее будет оставить их там, где они есть. НБ

6 Перенос KMail

Я только что приобрел ноутбук Toshiba и установил на него PCLinuxOS. На моем старом ноутбуке стоит OpenSUSE. Я хочу переместить свою почту на новый компьютер (на обоих я пользуюсь *KMail*), но не могу найти никакого варианта экспорта. Как переместить почту с одного компьютера на другой?

neilj

KMail хранит почту в `~/.kde/share/apps/kmail`, а настройки — в `~/.kde/share/config/kmailrc` и `~/.kde/share/config/emailedentities`. Просто скопируйте эти файлы на новый компьютер и убедитесь, что их владелец является владельцем также и остальных файлов KDE. Наверное, Вы захотите перенести еще и адресную книгу, тоже в `~/.kde4/share/apps`.

Узнать, где программа хранит свои настройки или данные, позволит простой и неспортивный путь. Откройте терминал и запустите `touch ~/timestamp`

Имя файла не имеет значения, это просто маркер текущего времени. Затем запустите свою программу, измените что-нибудь и выйдите из нее. Теперь вернитесь в терминал и скомануйте `find ~ -newer ~/timestamp`

Выведется список всех файлов и каталогов в Вашем домашнем каталоге, созданных или измененных с момента запуска той команды `touch`. Вывод может получиться довольно объемным, особенно если Вы запустили *KMail* и загрузили несколько писем, поэтому пропустите его через `less`, чтобы было проще найти искомое. Также

можно сузить область поиска – например, для программы KDE искать только в `.kde`.

```
find ~/.kde -newer ~/timestamp | less
```

Если Вы четко представляете, что ищете, пропустите результат через `grep`, и это сузит поиск:

```
find ~ -newer ~/timestamp | grep -i mail | less
```

7 Сканеры и VueScan

До последних версий Linux Mint и Ubuntu мой USB-сканер пленки Canoscan FS4000 работал в этих ОС через *VueScan*. Однако последний релиз *VueScan* (9.0.49) оказалось трудно даже установить, особенно на рабочем столе Unity.

При запуске этого *VueScan* мы получаем жуткое сообщение «сканер не обнаружен». В терминале *Isusb* выводит сканер Canon FS4000 в списке.

Я попробовал более старую модель Canoscan FS2700 в режиме SCSI, и она обнаружилась и работала (по счастью, у меня была установлена SCSI). У рабочего стола Unity в Ubuntu 11.04, похоже, проблемы с *Compiz*, и переключение на более простую тему *Classic* позволяет загрузить *VueScan* без проблем.

Как избавиться от проблемы обнаружения USB-сканера? Ведь без сканера пленки серьезно фотографу не обойтись.

Бернард Ривз [Bernard Reeves]

Только что попробовав *VueScan 9.0* со сканером Canon, я наткнулся на ту же проблему: сканер просто в упор не виден, даже если запускать программу от имени root.

У Вас есть несколько вариантов.

Так как сканер хорошо работает с версиями 8.5 и 8.6, можете их и держаться. Более старые версии программы все еще доступны на сайте *VueScan*, хотя было бы мудро сохранить копии TGZ-файлов для последующего использования.

Во-вторых, с целью известить *VueScan* о Вашем сканере можно задать три переменные окружения.

В переменную `SCANNER0` устанавливаете устройство, создаваемое в `/dev/` при подключении принтера, которое будет одним из списка `/dev/usbdevX.Y`, где X и Y – номера шины и устройства (их покажет вывод команды *Isusb*).

Она может изменяться при каждом подключении сканера, но есть способ это обойти.

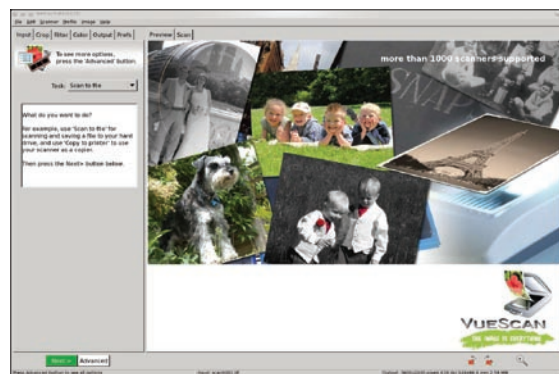
Имена двух других переменных – `SCANVID0` и `SCANPID0`, в них записываются идентификатор производителя (Vendor ID) и продукта (Product ID) соответственно. Эти данные также присутствуют в выводе *Isusb*.

Вот типичный вывод этой команды:

```
Bus 006 Device 003: ID 04a9:221c Canon, Inc. CanoScan
```

Идентификаторы производителя и продукта – два шестнадцатеричных числа, разделенных двоеточием. В данном случае переменные окружения можно задать командами

```
export SCANNER0="/dev/usbdev6.3"
export SCANVID0="04a9"
export SCANPID0="221c"
```



VueScan – хороший пакет, и, как с любым коммерческим ПО, при неполадках вам обязаны оказывать помощь.

Однако Вы вряд ли захотите все время их устанавливать – лучше сделать автоматическую ссылку на `/dev/scanner` для этого устройства, создав под пользователем root файл `/etc/udev/rules.d/10-scanner.rules` с такой строкой:

```
ATTR{idVendor}=="04a9",
ATTR{idProduct}=="221c",
GROUP=="scanner", MODE=="0660",
SYMLINK=="scanner"
```

Замените ID-коды подходящими для Вашего сканера. Отключите и подключите сканер, и должна появиться символическая ссылка `/dev/scanner` на созданный файл `/dev/usbdev`. Также проверится, что файл доступен для чтения всем членам группы `scanner`, чтобы ПО для работы со сканером не нужно было запускать от имени root, если Ваш пользователь входит в группу `scanner`. Теперь можно задать эти переменные навсегда, так как



Часто задаваемые вопросы

Мобильный широкополосный доступ

» Все вокруг болтают про мобильный широкополосный доступ. Это типа ADSL для мобильного телефона?

Вроде того. Здесь используется мобильная сеть 3G, но технология называется не ADSL, а HSPA (High Speed Packet Access – высокоскоростной пакетный доступ), она предназначена для мобильных устройств и используется в смартфонах, которым нужно постоянное (или по крайней мере частое) подключение к Интернету.

» Он дает скорость широкополосного доступа?

Нет, он не широкополосный в исконном смысле этого слова (как и некоторые медленные проводные соединения), но достаточно быстр для

важных задач вроде чтения почты и просмотр видеороликов на YouTube. Различные фирмы предлагают различные скорости доступа, но все они сильно зависят от уровня сигнала и других параметров.

» Как мне его получить?

Вам нужен широкополосный модем и договор с оператором мобильной связи. В большинстве договоров модем предоставляется, но можно воспользоваться и своим. Операторы чаще всего используют модемы в виде USB-брелков – в них вставляется SIM-карта для идентификации клиента.

» И сколько это стоит?

Как и с большинством таких вещей, цена зависит от поставщика, сро-

ка договора и объема данных. В Великобритании за 15 фунтов можно приобрести около 3 ГБ интернет-трафика в месяц.

» Каков срок действия договора?

От 1 до 18 месяцев; также предоставляются сервисы с оплатой за фактическое потребление трафика. Более длительные договоры предусматривают бесплатный модем.

» Хорошо ли это работает с Linux?

Да неплохо. Все компании предоставляют модемы одного и того же производителя, Huawei, а драйверы для него есть в ядре Linux, и он должен «просто работать». Нужно настроить модемное соединение с помощью программы вроде KPPP или Gnome-PPP, выбрав

в качестве устройства модема `/dev/ttyUSB0`. Так как для аутентификации используется SIM-карта, имя пользователя и пароль могут быть любыми (хотя в программе нужно их указывать).

» Играет ли роль используемый дистрибутив?

Стандартные настройки PPP должны работать в любом дистрибутиве. У дистрибутивов, использующих *Network Manager*, есть преимущество: он теперь поддерживает 3G-модемы. В таком дистрибутиве после вставки модема в компьютер появляется окошко с запросом о провайдере, и все настройки будут заданы за вас. Затем можно включить и отключать модем через меню *Network Manager*.

БОЛЬШОЙ ВОПРОС Как восстановить данные на жестком диске?

В Я здорово сглупил: устанавливая новый дистрибутив и задавая настройки изменения существующего раздела и установки с двойной загрузкой, выбрал не тот вариант, и установщик стал форматировать весь диск. Поняв, что натворил, я вырубил компьютер и заменил жесткий диск запасным. Как спасти данные с диска? Перспектива повторной установки Linux меня не очень огорчает, но жаль терять множество фотографий и музыки.

Патрик Уилсон (Patrick Wilson)

О Вы правильно сделали, что вынули диск. Если на него ничего не записывалось после перераспределения разделов, есть вероятность, что Ваши данные не только нетронуты, но и доступны. При создании новой схемы разделов на самом деле всего лишь перезаписывается таблица разделов в начале диска. Даже если Вы начали форматировать файловые системы на новых разделах, Вы пока не трогали области, содержавшие файлы (если граница между двумя разделами не попала в область диска, где прежде находился один из Ваших файлов).

Перезаписанную таблицу разделов обычно можно восстановить утилитой **testdisk**, включенной во многие Live CD. Загрузитесь с Live CD (мой любимый – *System Rescue CD*, <http://www.sysresccd.org>) и запустите **testdisk** из командной строки следующим образом:

```
testdisk /dev/sda
```

Выберите диск и тип таблицы разделов. Почти всегда это Intel, хотя в некоторых более новых системах бывает EFI. Выберите Analyse [Анализировать] в следующем окне, и **testdisk** покажет текущую структуру раздела. Первый шаг – воспользоваться опцией Backup [Резервная копия], чтобы сохранить текущую схему раздела. Ее нужно где-то сохранить, но ни в коем случае не на диск, который Вы пытаетесь восстановить – запишите ее на USB-флэшку. Quick Search [Быстрый поиск] обычно выдает ту же самую информацию, но затем Вы сможете выбрать Deeper Search [Углубленный поиск] для поиска дальнейших разделов. Этот этап может потребовать времени, поскольку каждый блок диска исследуется на наличие маркеров начала раздела.

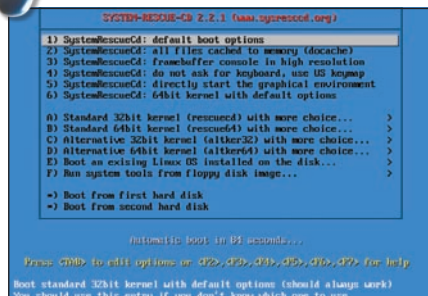
Отсюда часто можно сбросить таблицу разделов в предыдущее состояние, и тогда файлы окажутся там, где их оставили. Даже если раздел нельзя восстановить, **testdisk** часто может обнаружить файлы, которые прежде были на диске, хотя для этого потребуется второй диск, куда они будут скопированы. Это предпочтительный метод, так как он не включает запись на исходный диск и нет шансов ухудшить ситуацию.

Если разделы или структура файловой системы повреждены слишком сильно, все еще есть надежда восстановить файлы с помощью **photorec**. Несмотря на название, с помощью этой программы можно восстанавливать файлы любых типов с дисков любых типов.

Недостаток этого метода в том, что так как недоступны таблицы файловой системы, то нет метаданных, которые нужно восстанавливать. Но для фотографий это не проблема, поскольку метаданные у них хранятся в разделе EXIF самого файла. Аналогично, для музыкальных файлов вся необходимая информация хранится в тэгах ID3.

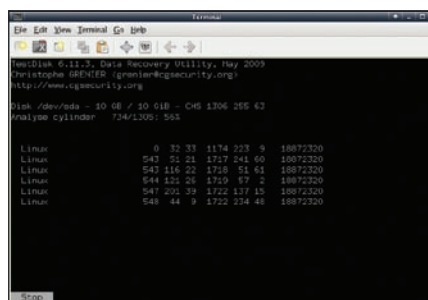


Шаг за шагом: Спасаем систему



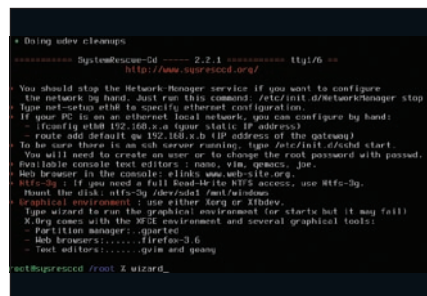
1 System Rescue CD

У каждого пользователя компьютера должна быть копия *System Rescue CD* или похожего Live CD. С его помощью даже можно восстановить данные Windows.



4 Доступные разделы

После глубокого сканирования **testdisk** покажет все доступные разделы. Может потребоваться взглянуть на каждый раздел, чтобы выбрать для восстановления нужный.

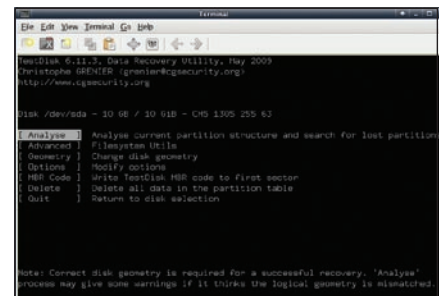


2 Клавиатура или мышь?

По умолчанию *System Rescue CD* работает в текстовой консоли, и это прекрасно подходит для запуска **testdisk**. Если вы предпочитаете графический интерфейс, запустите мастер настройки X.

5 Что делать

У каждого раздела есть несколько опций; первым делом нажмите P для вывода списка файлов. Так вы убедитесь, что действительно выбрали нужный раздел.



3 Опции testdisk

После запуска **testdisk** и выбора нужного диска вы видите несколько опций. Чтобы восстановить поврежденный диск, начните с Analyse.

6 Восстановим файлы

Даже если вы не можете или не хотите восстанавливать раздел целиком, обычно можно просмотреть его файловую структуру и скопировать данные на другой диск. **lsx**



Лучшее в мире новое ПО
с открытым кодом

LXF HotPicks



Ник Вейч

Ника однажды укусил радиоактивный менеджер скачиваний, и с тех пор у него болезненная страсть к экспериментам с установкой программ. И дурное пищеварение.

Amarok » Archipel » IPython » Takeoff » Rowscope » GPodder » Minitube » CKhet » Cubosphere » Usermin » Tiny Core Linux

Музыкальный плеер

Amarok

Версия 2.4.3 Сайт <http://amarok.kde.org>

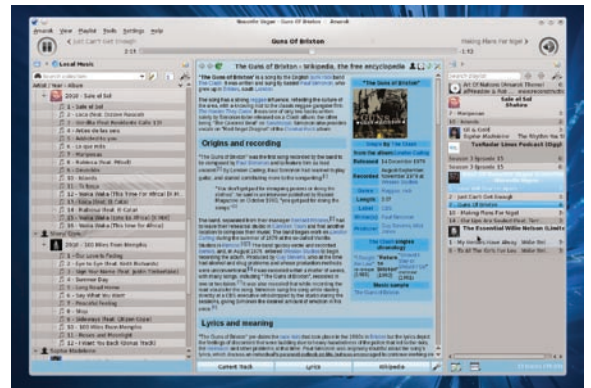
Есть музыкальные плееры, а есть *Amarok*. Возможно, в прошлом мы слегка пренебрежительно относились к этой «улучшенной» версии одного из самых мощных аудиоинструментов в арсенале Linux (а так и было, раз мы включили *Clementine* в выпуск LXF145), но он слишком хорош, чтобы его игнорировать, несмотря даже на причудливые одежды, в которые он ныне облачен. На самом деле, это как раз одно из изменений. Пользовательский интерфейс стал чуть менее перегруженным благодаря реконструкции, заменившей строку состояния, подправившей меню, и лицевой фурнитуре.

Если раньше вы не имели дела с *Amarok* (мало ли, вдруг вы выросли на Ubuntu

и заранее напуганы всем KDE-арсеналом), готовьтесь пережить потрясение, которое вы непременно испытаете от всех его музыкальных чудес.

Естественно, тут имеется все, что полагается по стандарту, то есть управление музыкальным каталогом, работа с устройствами МРЗ-плеера, поиск изображений и создание списков воспроизведения; но все это *Amarok* делает осмысленно и с апломбом, и снабжен также

«Если вы не имели с ним дела, готовьтесь к потрясению.»



» Откуда бы вы ни заполучили музыку или аудио, *Amarok* с ней справится.

такими функциями, как поиск в Сети слов песен и информации по треку. Помимо локальных файлов, он также может управлять подкастами и прочими интернет-ресурсами типа Last.fm, или настроиться на местный источник DAAP или UPnP. Фактически, откуда бы вы ни заполучили музыку, *Amarok* ее воспроизведет.

Правда, проблемой при попытке скомпилировать его из исходника бывает зависимость от *libmysqld*, встраиваемой версии *MySQL*, которая не поставляется с основным кодом *MySQL* (5.0), и вам, скорее всего, придется устанавливать ее отдельно, поскольку большинство современных дистрибутивов уже работают на 5.0. Можно схитрить: использовать для поиска нужных пакетов требования по сборке от предыдущей версии. В Debian/Ubuntu это будет

```
apt-get build-dep amarok
```

а в Fedora можете использовать

```
yum-builddep amarok
```

Затем продолжайте компиляцию обычным способом. *Amarok* имеется практически во всех известных репозиториях; если вы хотите сэкономить полчаса, там, без сомнения, скоро появится и обновленная версия.

Исследуем интерфейс Amarok

Список воспроизведения

Стандартная раскладка включает вашу музыкальную коллекцию в виде древовидного отображения на левой панели – можно выбрать и другие источники, например, Интернет.

Обложка диска Cover artwork

Обложка диска считается с коллекции или, если вы захотите, ищется в Интернет.

Слова песен

Слова песен также могут быть найдены и отображены, и при желании они могут прокручиваться по мере звучания трека.



Интерфейс

Пользовательский интерфейс был отлажен, в том числе строку состояния заменили на более подходящее к ситуации уведомление.

Настройки

Как обычно, имеется масса страниц настроек, по которым можно ползать.

Управление VM

Archipel

Версия beta 3 Сайт <http://archipelproject.org/>

Виртуальные машины (VM) – это лучшее, что досталось человечеству после, кхе, настоящих машин. Они умудрились преодолеть свою неуверенность и неуклюжесть, особенно при тестировании или разработке программ, потому что целые системы можно создать из файла `config`, и они будут готовы к работе прежде, чем вы придумаете, что же с ними делать.

Linux благословлен массой технологичных VM, причем большая их часть идет под свободными лицензиями.

И хотя по большей части они концентрировались на эмуляции работы оборудования, все больше внимания в последнее время обращается на управление всеми этими виртуальными машинами; и *Archipel* – еще один инструмент, способный помочь вам в этой области.

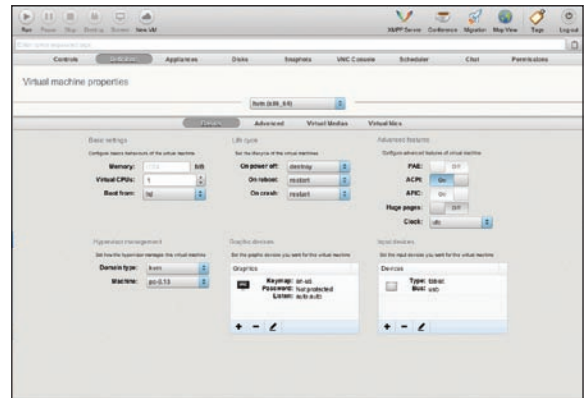
Работая с программами для виртуальных машин на основе *libvirt* (среды которых – *KVM/ qemu*, *VmWare*, *Xen*,

VirtualBox...), он предоставляет унифицированный полнофункциональный интерфейс для управления и мониторинга всей толпы ваших машин.

Кстати, мы уже говорили, что эти машины не обязаны быть локальными? *Archipel* использует протокол обмена быстрыми сообщениями Jabber с некоторыми изменениями для удаленного общения с виртуальными машинами, где бы они ни находились.

Теневой стороной всего этого является настройка.

Разобраться в самом *Archipel* довольно просто, но вот настроить его, даже для простого использования на одной системе, куда сложнее, чем хотелось бы: настройка



► *Archipel* позволяет дистанционно управлять виртуальными машинами через полнофункциональное web-приложение.

включает часть компиляции вручную и огромные трудозатраты на получение файла рабочей конфигурации. В документации предполагается, что вы используете систему на базе Debian, что не всегда верно, а если вы хотите изменить настройки в клиенте по умолчанию (все они созданы на JavaScript), придется скачать ряд инструментов разработки для рекомпиляции.

Однако не надо забывать, что это – бета-релиз, и если вы его все-таки запустите, *Archipel* весьма впечатляет.

«Linux благословлен
массой технологий
виртуальных машин.»

Интерактивная оболочка Python

IPython

Версия 0.11 Сайт <http://ipython.org/>

В глубине души, Python – язык интерактивный. Большинство уроков по Python, опубликованных в *LXF*, начинаются с экспериментов в интерактивной оболочке Python: просто набираете в командной строке `python` – и вперед.

Но для некоторых одной этой простоты мало – истинно интерактивная среда должна иметь больше функций и полезных дополнений, чтобы облегчить жизнь экспериментаторов.

IPython как раз и является такой средой – с виду она ведет себя, как стандартная сессия Python, запущенная из оболочки, но очень скоро вы заметите разницу.

Лидером по экономии времени является, наверное, автозаполнение вкладок, если вам нравится их использовать. Оно работает на удивление хорошо и учитывает действующий синтаксис, модули и т.п., так что вы получаете только правильные опции для набираемых команд. Добавьте к этому полную историю ввода и вывода, и у вас получится интерактивная версия

Python, которая в плане команд приближается к оболочке Bash.

Пускай некоторые высмеивают его быстрый интерфейс и чрезмерные интервалы, которые он применяет для красоты, однако вряд ли заслуживает осмеяния возможность вызвать команду оболочки, просто поставив ! перед ней.

IPython также доступен для различных версий Python, и вы можете наслаждаться им равно на 2.x и 3.x; только убедитесь, что он установлен с помощью правильного скрипта.

Пуристы могут и дальше полагать, что хорошее для Гвидо [Guido van Rossum – автор языка Python, – прим. ред.] хорошо и для всех нас, но тут дело в том, что зачастую привычка к чему-либо мешает нам

```

In [19]: ?
Out[19]:
In [20]: help
Out[20]: Type help() for interactive help, or help(object) for help about object.
In [21]: x=748983274583275489324
In [22]: help(x)
Out[22]:
In [23]: !free
total      used      free     shared    buffers     cached
Mem:    8152716  772644    426972      0      226688    2544228
-/+ buffers/cache: 4955816  3196908
Swap:   9797628      0      9797628

In [24]: import matplotlib
In [25]: help(matplotlib)
Out[25]:
In [26]: x - 1.0
Out[26]: 7.489832745832755e+20
In [27]: x**error
-----
TypeError                                 Traceback (most recent call last)
/home/evilnick/~ipython console> in <module>()
TypeError: unsupported operand type(s) for +: 'long' and 'str'
In [28]:
  
```

► Даже ваши хромые скрипты *IPython* сумеет превратить во впечатляюще-красочные.

вернуться к более старым и чуть более сложным вещам. Мы сейчас обленились, и избалованы всякими подпорками. Вот было бы здорово автозаполнять предложения, выбирая из списка классических фраз для завершения материала...

«Лидер по экономии
времени, наверное,
автозаполнение.»

Плеер подкастов

GPodder

Версия 2.18 Сайт <http://gpodder.org/>

Безусловно, есть очень хорошие подкасты, которые стоит послушать. Они посвящаются самым разным темам: среди них сделай-сам, массаж, классическая архитектура, игры, кулинария, гаджеты, и – да, конечно, даже Linux. (Вы, естественно, уже давно постоянный слушатель подкаста LXF на www.tuxradar.com.)

Загрузить уже транслируемый подкаст не слишком сложно, но если вы хотите, чтобы самые свежие выпуски постоянно появлялись на вашем MP3-плеере, придется обратиться к программе управления подкастами, и вот тут *Gpodder* почти всех победит (включая и приложение, которое мы сами написали в LXF134).

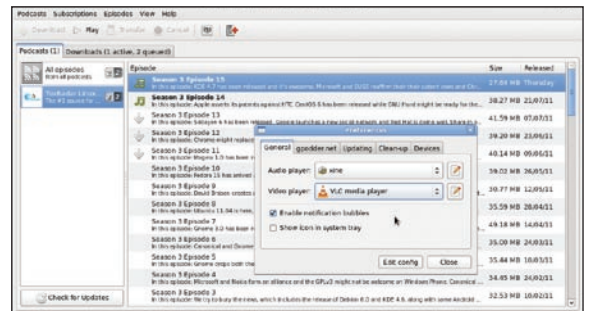
GPodder мастерски написан на Python/GTK и способен работать на разных платформах: вы даже сможете запустить его прямо на своем мобильном телефоне, а не передавать скачанные файлы. И хотя в основном упор сделан на простоту и удобство, функционал у *GPodder* весьма

неплохой. Он загружает значки каналов для подкастов и заполняет списки каналов любимыми комментариями и названиями каналов, поясняя вам, что именно вы получаете.

Он поддерживает каналы RSS и atom, и с равным успехом работает и с видео-, и с аудиоподкастами. Он поддерживает передачу на различные виды iPod (с помощью дополнительных программ), и может даже осуществлять передачу через Bluetooth (не рекомендуется для больших Ogg-файлов!).

В нем нет ничего особо зрелищного или чудесного, кроме редкостной способности делать свое дело – в вечно изменчивом мире свободного ПО это достижение порой недооценивают.

«У GPodder есть редкостная способность делать свое дело.»



➤ Теперь вы не пропустите подкасты, на которые подписаны – при всем желании.

GPodder активно развивается, и проблемы (обычно связанные с внешним миром, а не с самим *GPodder*) всегда выявляются быстро.

Его даже не нужно по-настоящему устанавливать. Если вы распакуете tar-архив и просто запустите `make test`, он запустится из этой директории, так что не познакомиться с ним будет неприятно. Сделайте это прямо сейчас!

Программа для просмотра YouTube

Minitube

Версия 1.5 Сайт <http://flavio.tordini.org/minitube>

Под силу ли одиночке изменить способ просмотра нами видео? Вероятно, нет; а вот Флавио Тордини [Flavio Tordini], невзирая на это, все равно решил рискнуть. Его простая-но-хитроумная идея заключается в создании отдельного плеера для просмотра видео с YouTube.

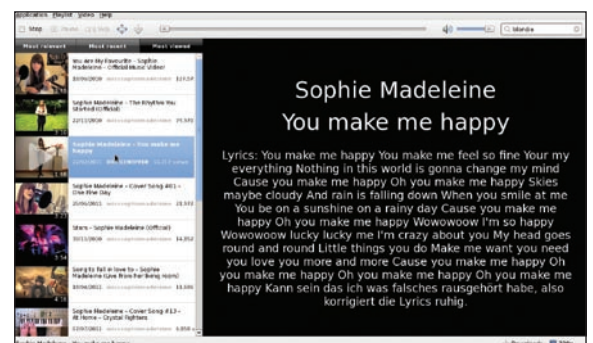
Вы можете сказать: а что, разве это не просто окно браузера? Не тут-то было! Ну, вообще-то отчасти да. Но поскольку *Minitube* не выглядит простым окном браузера, и в нем имеются функции, в большей степени нацеленные на зрителей, а не на пользователей Web, он этак мягко и тактично изменяет ваше восприятие происходящего. Кажется, что вы уже не просто щелкаете по ссылкам, чтобы посмотреть видео с котами, подлизывающимися к хозяевам, или с кусающимися друг друга малышами, а на самом деле смотрите каналы с, э-э, котами, подлизывающимися к хозяевам.

Нет, *Minitube* и вправду хорош. Он не ускоряет просмотр Youtube, и все так же приходится томиться ожиданием, пока файлы загрузятся в буфер и т.д., но зато объединяет все нужные вам функции управления в симпатичном интерфейсе.

Несколько озадачивает помещение кнопок управления в верхнюю часть экрана, а не в нижнюю часть окна воспроизведения видео, но нельзя же получить все и сразу.

Minitube довольно легко компилируется из исходника. Будучи написан на прямодушном C++ для системы компиляции *Qtmake*, он не доставит проблем тем, кто в состоянии управиться с парой-тройкой команд.

«Он мягко изменяет ваше восприятие происходящего.»



➤ Смотрите свои любимые каналы, не отвлекаясь на web.

Убедитесь в наличии у вас *Qt 4.5* или выше, и введите:

```
qmake && make
```

Надо упомянуть одну вещь: *Minitube*, возможно, работать не будет. Ну, сейчас-то он работает, но у YouTube есть привычка постоянно вводить изменения в сайт, и это приводило к сбоям в предыдущих версиях программы.

Если он не будет работать, автор обычно очень быстро принимает меры, так что загляните на сайт в поисках обновлений.

HotGames Развлекательные приложения

Стратегия

CKhet

Версия 0.1.6 Сайт <http://mbays.freeshell.org/ckhet/>

Древние египтяне, которым, по слухам, содействовали инопланетные технологии, оставленные в форме пирамид, ничто не любил больше, чем разрушительную игру с лазерным брелком при участии зеркал и древних статуй. По-видимому. Именно такова, кажется, основа популярной настольной игры *Khet*, которая вовлекает настоящих лазерных сфинксов (убедитесь сами на <http://www.khet.com>).

Компьютеризированная версия использует не лазеры, а curses, и почти так же хороша. Только вам придется вообразить себе все эти любовно отделанные пластиковые фишки, представляющие Анубиса и К°, созерцая цветных ASCII-персонажей на экране.

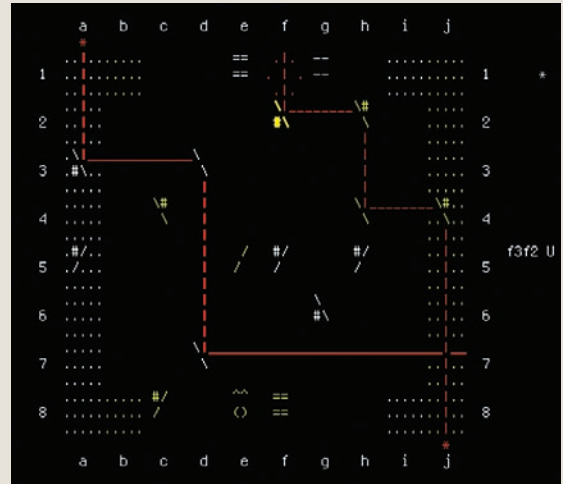
Но игры – это не одна лишь графика. *CKhet* достаточно проста: передвигайте фишки вокруг стола и в конце своего хо-

да палите из лазера (в смысле, цветного символа '-'), чтобы накрыть другого игрока. На самом деле, это довольно притягательно: как шахматы, но с оружием.

Данная версия переделана на Python, так что она в высшей степени способна к изменению. Область, с которой надо бы поработать – интеллект компьютерных игроков. Можно играть и против реальных игроков, включив сетевой режим.

По сути, игра работает в классическом режиме сервер–клиент, и вам нужно открыть два терминала и запустить программы в каждом из них. Устанавливать ее не нужно – просто запустите

«Довольно притягательно: как шахматы, но с оружием.»



› Похоже, что средневековые правила боя в основном полагались на точное расписание.

ее из директории, куда вы ее распаковали, введя

```
./ckhet.py -r basic
```

и в другом терминале:

```
./ckhet.py localhost
```

Насчет зрения зависимостей особо беспокоиться нечего: все, что вам потребуется – Ncurses.

Даже лучшие игры бывают довольно просты; а эта, пусть и в своем скромном великолепии ASCII, весьма интригует и затягивает.

Игра в шарик

Cubosphere

Версия beta Сайт <http://sourceforge.net>

У игры в шарик долгая история. Есть что-то чарующе-первобытное в законах физики, заставляющих катиться шарик, скованный и силой тяжести, и инерцией.

Но одно дело – гонять шарик по лабиринту, и совсем другое – управлять ими в мире *Cubosphere* со его прихотливой гравитацией.

Ваша цель – вырваться из лабиринта, но для этого нужно, как правило, собирать ключи от хитроумных локаций, избегая по дороге падений в ловушки. И все это становится еще сложнее, благодаря множеству других «функций», возникающих именно тогда, когда вы этого совершенно не ждете, и изменчивой по своей природе силе тяжести в трехмерной среде, в которой не разберешь, где верх, а где низ.

Если вы сочтете вид игры знакомым, вам, вероятно, полезно будет узнать, что на нее сильно «повлияла» *Kula World*, большой хит PlayStation лет двенадцать назад – она скорее напоминает ее, чем более старые игры типа *Marble Madness*.

В игре более 200 уровней (у нас не было времени пройти все), но если они вам надоедают или захочется сыграть злую шутку с друзьями, всегда можно создать свой собственный.

Редактор уровней – одна из самых впечатляющих вещей в игре: он прост в использовании, идет с обширной доку-

«Ваша цель – вырваться из лабиринта, собирая ключи.»



› Уровень с Альпийской темой включает телепорты, переключатели и, хм, очки...

ментацией в формате PDF и хорошо продуман, и вы очень скоро приметесь клепать собственные изуверские уровни.

Завораживающая графика игры требует наличия 3D-ускорения, и вам понадобится обычный набор GLU, OpenGL и т.д. для компиляции из исходника. Бинарников пока нет, поскольку игра находится на стадии бета-тестирования, но мы уведомим вас, когда будет сделан полный релиз.

Инструмент администратора

Usermin

Версия 1.48 Сайт <http://www.webmin.com/usermin.html>

Usermin уже красовался на страницах HotPicks, но с тех пор прошло уже больше года, и мы снова включаем его в наш материал – по двум причинам. Во-первых, он явно стал еще лучше. Помимо обычной отладки, изменен раздел администрирования электронной почты – предотвращена возможность неприятных повторов электронной почты и добавлены новые опции уведомления. Естественно, обновлены многие модули расширения, и есть масса дополнений от третьих сторон.

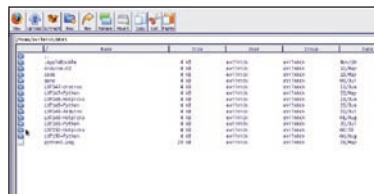
Usermin – часть семейства сетевых инструментов администрирования Webmin. В основном он существует потому, что, по мнению системных администраторов, Webmin весьма полезен для управления удаленными серверами. Настолько полезен, что все рвутся попробовать.

Usermin – нечто вроде подмножества функций, позволяющего «обычному пользователю» получить доступ к тем вещам, которые обычные пользователи должны уметь делать локально. Фактически,

Webmin полезен настолько, что вам даже не нужен удаленный компьютер, чтобы им пользоваться.

Унифицирующие администраторский интерфейс для множества вариаций Linux, Webmin и Usermin – действительно полезные и удобные инструменты для тех, кто отчаянно пытается запомнить, где именно в каждом дистрибутиве хранятся важнейшие части.

Вторая причина? Мы приняли во внимание тот факт, что далеко не каждый читает выпуски LXF все подряд, и есть некая вероятность, что вы пропустили эту приятную мелочь. Если это так, пожалуйста, не делайте в будущем подобных глупостей. И попробуйте Usermin, он превосходит.



➤ Usermin поддерживает NetBSD, FreeBSD и множество других разновидностей Unix.



Дистрибутив Linux

Tiny Core Linux

Версия 3.7 Сайт <http://distro.ibiblio.org/tinycorelinux>

Это полностью рабочая версия Linux, которая невероятным образом вместились всего в 10 МБ. Ну да, это не самая впечатляющая версия Linux, но это определенно Linux – он включает основные команды оболочки (благодаря BusyBox) и даже рабочий X-сервер.

Tiny Core основан на серии более компактных ядер Linux 2.6 и жестко втиснут в заданные рамки. В итоге, при желании работать с более сложными программами вам придется многое в него добавить.

Если вам нужен мини-дистрибутив со всеми обычными атрибутами рабочего стола, лучше, наверное, начать с чего-то другого, но для множества специфических областей применения, помимо простого любопытства, Tiny Core являет собой вполне приличный полигон. В основном он удобен для установки на встраиваемые устройства, но вполне сгодится для реанимации старого оборудования или

создания системы Linux ради какой-то одной специфической цели.

Tiny Core загружает все в ОЗУ (у вас явно останется пара свободных МБ) и работает по-настоящему быстро. Дополнительные программы тоже можно загрузить в ОЗУ или запустить из их обычного места хранения, и, конечно же, можно добавлять к ядру драйверы оборудования.

У Tiny Core активное сообщество, и доступно множество подсказок и документации. Есть немало «пакетов новичка» для определенных задач – вы без проблем выкатите собственную версию.



➤ Пускай он некакист, но зато занимает меньше места, чем графический файл с высоким разрешением.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

StrongSwan 4.5.3

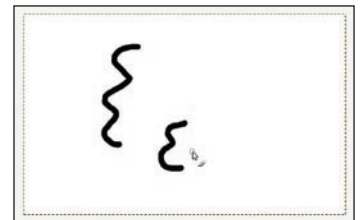
IPsec, всерьез подогнанный под Linux и Android, если вам нужен VPN на телефоне.
<http://www.strongswan.org>

fio 1.5.6

Впечатляющий стресс-тестер для операций ввода/вывода.
<http://git.kernel.dk/?p=fio.git;a=summary>

Shutter 0.87.3

Отличная программа для снимков с экрана теперь делает правильные снимки курсора.
<http://shutter-project.org>



➤ Делает снимки. С экрана. А теперь и курсор правильно отображает.

amforth4.5

Устали от Arduino? Перепрограммируйте свои чипы Atmel и работайте с интерактивным FORTH!
<http://amforth.sourceforge.net>

echinus 0.4.6

Если вы хотите, чтобы ваш легковесный менеджер окон был едва заметен.
<http://plhk.ru>

Tea 30.1.0

Новая версия вечно популярного текстового редактора на базе Qt.
<http://tea-editor.sourceforge.net>

Parted Magic 6.4

Полный дистрибутив для работы с разделами диска – теперь с ядром 3.0.
<http://partedmagic.com/doku.php>

eko 0.3.0

Действительно простой, но эффективный редактор сэмплов.
<http://eko.sourceforge.net>



➤ Судя по форме кривой, смахивает на кашель Майка.

На диске

Дистрибутивы, приложения, игры, подкасты и многое другое...



Прочтите всю информацию, необходимую для использования DVD! Если вы новичок в Linux, откройте файл index.html на диске и перейдите в раздел Справка, где вы найдете руководства по вопросам, перечисленным справа:

Майк Сондерс, редактор диска
mike.saunders@futurenet.com

- » Что такое Linux?
- » Что такое дистрибутив?
- » Загрузка ПК с DVD
- » Разбиение жесткого диска на разделы
- » Навигация по файловой системе
- » Использование командной строки
- » Установка программ

Дистрибутив Linux

Chakra 2011.09m3

Только что вышел KDE 4.7, как вы поняли из нашего обзора на стр. 10, и мы решили выдать вам его на пробу. Но мы не просто запихнули на диск исходный код, а нашли дистрибутив, где он имеется в прекомпилированном виде. И, по счастью, им оказался именно тот, который нам не так давно рекомендовали – Chakra Linux.

Чакра, в традиции индийских йогов, это из жизни энергетических каналов человеческого тела; но не бойтесь, вам не придется завязаться немислимимым узлом, чтобы в нем работать. Он запускается в *VirtualBox*. Философия команды Chakra – KISS: Keep It Simple, Stupid [Будь проще, придурок]. Или, в более вежливой форме, Keep It Short and Simple [Коротко и Просто]. По-

этому он не пытается стать всем для всех и каждого, зато позволяет отлично поработать с KDE и Qt, сразу и без проволочек. В первую очередь он предназначался для новичков, но не обижены и «технари».

Чтобы запустить Chakra, загрузите свой компьютер с *LXF DVD* и выберите его в меню загрузки. Если во время загрузки на экране появятся визуальные артефак-

Если увиденное придется вам по душе и вы решите постоянно пользоваться Chakra, можете запустить программу установки на жесткий диск через значок на рабочем столе. Однако помните, что это альфа-релиз дистрибутива, и мы не рекомендуем его для рабочих машин.

Это неплохой способ позабавиться с самыми свежими программами, но на машинах, для которых главное – надежность, мы рекомендуем лучше установить CentOS (см. через страницу).

Предупреждение Chakra, что программа установки «может съесть вашего хомячка» должно вам все прояснить...

Но все же программа установки довольно проста в использовании, и по окончании процедуры можете перезагрузиться и извлечь диск. Более подробную информацию см. на сайте www.chakra-project.org, а помощь от других пользователей вы получите на форумах: www.chakra-project.org/bbs/.

«Это альфа-релиз, мы не рекомендуем его для рабочих машин.»

ты, или вообще не появится изображения, перезапустите машину и попробуйте выбрать опцию *Old Display* в меню загрузки. Через несколько мгновений перед вами появится рабочий стол, и вы сможете начать знакомство с имеющимися программами через меню внизу слева.

Важно ВНИМАНИЕ!

» Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

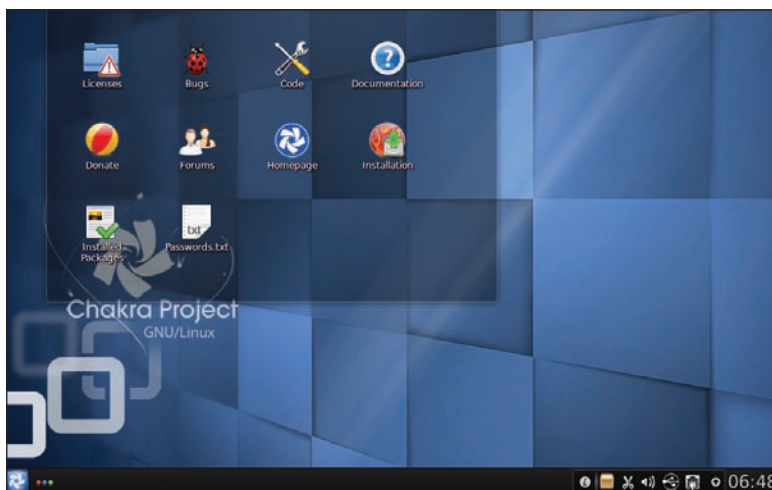
Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.



» Бум-шакалака Chakra! — быстрый, рациональный дистрибутив с самым свежим KDE. Ну как его не полюбить?

Дистрибутив Linux

Ubuntu 11.10 Alpha 3

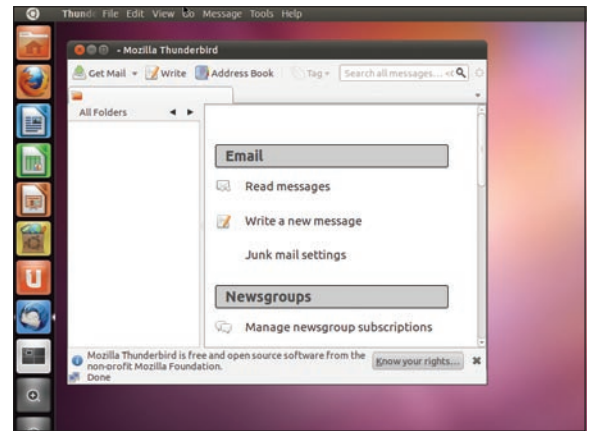
Грядет новый релиз Ubuntu! Когда вы будете читать этот материал, до него останется всего ничего. Мы всегда с нетерпением и трепетом ожидаем новой версии самого популярного в мире дистрибутива, но за этой версией следили особенно внимательно. В конце концов, новый интерфейс Unity не везде был принят одинаково хорошо, и для Сапописа это возможность исправить недочеты и ошибки, а также снять ограничения, вызвавшие столько жалоб.

Это третий альфа-релиз будущего Ubuntu 11.10 (он же – Oneiric Ocelot), предназначенный не для повседневной работы, а только для тестирования и создания представления о том, что будет в дистрибутиве. Вы найдете его на **LXF DVD** в разделе Distros/Ubuntu; он идет в виде ISO-образа, который можно записать на CD-R и загрузиться с него. Помните, что вы должны

записывать его как ISO-образ, а не просто копировать .iso на пустой диск – если вы не уверены в том, как это сделать, взгляните в документацию вашей программы для записи дисков.

Как обычно, Ubuntu работает прямо с CD, и вам не придется устанавливать его, если вы сами этого не захотите (получите солидный прирост скорости). Но вы можете сразу познакомиться с новыми функциями. Вот на что стоит обратить особое внимание: появился новый экран приглашения в виде *Lightdm*, взамен *Gdm*. *Thunderbird* сменил *Evolution* в качестве почтового клиента по умолчанию, а инструмент резервного копирования *Deja Dup* теперь включен в стандартный образ CD.

Под капотом, ядро достигло версии 3.0, а версии Unity 2D и 3D имеют больше общего кода. В общем и целом, изменения довольно значительны. Одним они попра-



➤ Прощай, *Evolution!*
Уступи дорогу *Thunderbird*.

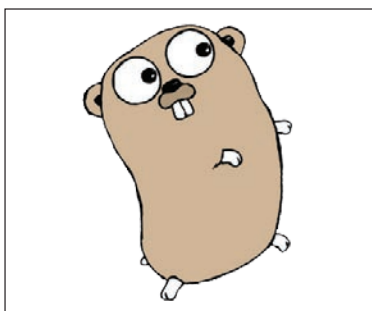
вятся, другие их возненавидят; но, надемся, к концу процесса разработки все постепенно войдет в свое русло и мы получим хороший, прочный релиз, который не обманет наших ожиданий.

Другие программы

Инструменты разработки, руководства и подкасты...

По возможности, мы стараемся помещать на диск программы, о которых рассказываем в журнале. Например, после чтения интервью с Робом Пайком на стр. 30 вы, наверное, рветесь познакомиться с Google Go. В разделе Development на диске помещена свободная реализация этого интересного языка и курс обучения по нему, который проводит лично Роб Пайк.

Также в разделе Development находится Phing, и это – вы уже догадались – один из акронимов. Расшифровывается он так:



➤ Суслик Google Go глумится над тормозными интерпретируемыми языками.

PHing Is Not Gnu Make (PHing – Не Gnu Make), и предоставляет систему компиляции, основанную на *Apache Ant*. Он впечатляет своей гибкостью и поддерживает трансформации файлов, интерактивные сборки, исполнение команд SQL, операции CVS и другие интересные функции.

Для тех, кто только начал осваивать мир Linux, у нас есть подборка руководств. Не бойтесь – это вовсе не долгий и занудный процесс, а сжатые инструкции, которые помогут вам освоиться. Откройте **index.html** на диске и перейдите в раздел Help/New to Linux.

И последними по очереди, но не по значимости идут самые свежие подкасты Tux-Radar. Наш читатель Heiowge сказал нам, что узнал о них только сейчас, а это значит, что мы не уделяем должного внимания их рекламе. Каждые две недели влиятельные особы из Linux Format берут микрофоны и экземпляр *Audacity* и обсуждают важнейшие проблемы Linux и свободного ПО. Зайдите в раздел Magazine/Podcast, чтобы прослушать самые последние выпуски, и взгляните на www.tuxradar.com/podcast, где найдутся и другие.

И это еще не все!

Раздел Ретростальгии

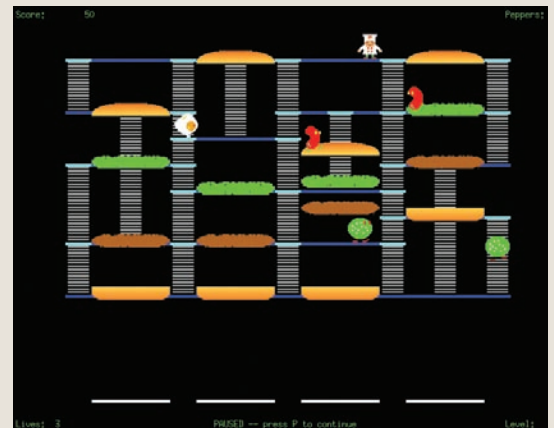
В этом месяце у нас тут немного ретро-темы в игровом разделе. Первой идет *BurgerSpace*, «видеоигра-крушитель гамбургеров» – это клон *BurgerTime*, аркады 1982 года. На диске – ее исходный код и RPM-пакеты

Затем идет *LGeneral*, пошаговая стратегия, навеянная *Panzer General*. Вы управляете танками, самолетами и солдатами на поле боя с целью победить соперника.

А вот *Teeworlds* описывается разработчиками как «переходная стадия от *Quake* к *Worms*» –

интригующая концепция. Вы управляете бандой мультяшных зверюшек,двигающихся по 2D-уровню, размахивающих крючками и применяющими все виды оружия. Главное отличие от *Worms* в том, что все происходит в реальном времени, не переходя в пошаговый режим.

И, наконец, *Tyrant*, Roguelike-подобная [roguelike] игра, написанная на Java. Если слово «рогалик» для вас значит «выпечка», срывайте погony хакера и идите на <http://en.wikipedia.org/wiki/Roguelike>.



➤ Разгул коровьего бешенства в *BurgerSpace*.

На диске

Встречайте виртуально непобедимый дистрибутив Linux...

Дистрибутив Linux

CentOS 6

Есть надежные, есть супер-надежные, и есть CentOS. Справедливым будет сказать, что большинство известных дистрибутивов достаточно солидны – полный сбой системы случается крайне редко, а для многих вообще исключен – но для очень серьезных ситуаций нужен и серьезный дистрибутив. Вы, наверное, ищете нечто вроде Red Hat Enterprise Linux (RHEL)... но он довольно дорого стоит. Однако к счастью, и мы благодарны за это Red Hat, компания предоставляет доступ практически ко всем своим работам в формате SRPM (source RPM), так что каждый может сделать свою версию.

Именно так и возник CentOS. По сути, это RHEL, но бренд и прочие торговые

марки удалены из нее независимым сообществом разработчиков. Конечный результат – исключительно функциональный, хорошо протестированный дистрибутив, который будет получать пакеты обновлений еще долгие, долгие годы. Он отлично подходит для рабочих машин в бизнесе, являясь также прекрасным выбором и для домашних ПК, да и для серверов, если вы хотите, чтобы машина спокойно работала, не прерывая рабочего процесса получением обновлений.

Если вы подумывали поискать работу, связанную с Linux, вам определенно стоит познакомиться с CentOS. RHEL – стандарт основной платформы Linux многих компаний, а поскольку CentOS практически идентичен, все навыки и умения, приобретенные вами в этом дистрибутиве, вполне применимы к его коммерческому эквиваленту. Упражнения в CentOS также могут стать хорошей подготовкой к сертификации Red Hat, даваемой курсами RHCE и RHCT.

Вы можете запустить CentOS 6, который создан из исходного кода RHEL 6, прямо с LXFVD: просто установите диск в дисковод, перезагрузитесь, и вы увидите меню. (Если меню не появилось, ваш компьютер, видимо, настроен на загрузку с жесткого диска, а не с оптического. В таком случае вам стоит справиться с документацией вашего компьютера и изменить



После установки и первой загрузки CentOS настройте учетную запись обычного пользователя, для входа в систему.

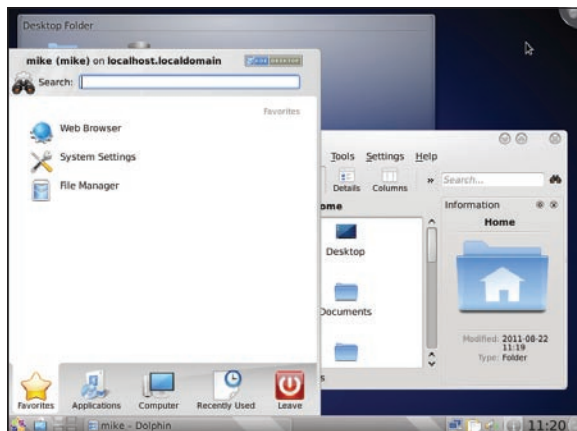
порядок загрузки в BIOS.) Рекомендуемые системные требования:

- » 1 ГГц x86 CPU
- » 512 МБ ОЗУ
- » 10 Гб места на жестком диске.

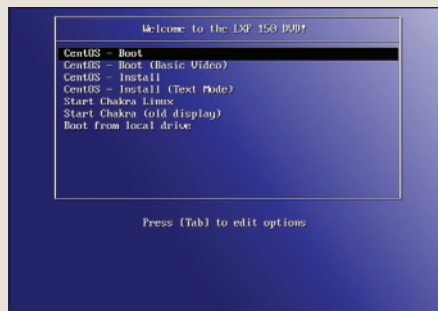
Этого достаточно для обычного настольного ПК или сервера, но для более требовательной станции разработчика мы бы рекомендовали 2-ГГц чип и как минимум вдвое больше ОЗУ. И хотя версия на LXFVD 32-битная, она с успехом должна работать и на 64-битных машинах..

Примечание для новичков в Linux: если у вас на жестком диске уже стоит ОС, скажем, Windows, можно установить CentOS рядом с ней, сжав раздел Windows в процессе установки. Проблем тут обычно не бывает, но все же рекомендуем сначала сделать резервные копии данных Windows! И – следуйте инструкциям внизу.

» Рабочий стол по умолчанию – Gnome. Для установки KDE выйдите из системы, щелкните по своему имени в логине и выберите KDE в панели внизу экрана.



Шаг за шагом: Устанавливаем CentOS 6



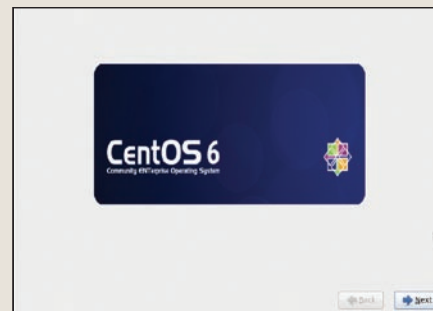
1 Загрузка

Загрузите свой ПК с LXFVD и нажмите на Enter в меню. (Если при загрузке возникли проблемы, перезагрузитесь и попробуйте опцию Basic Video.)



2 Рабочий стол

Вы автоматически попадете на рабочий стол. Исследовав имеющиеся программы, дважды щелкните по значку Install на рабочем столе.



3 Программа установки

Программа установки загрузится и задаст вам несколько вопросов. Вам нужно будет задать пароль для root (admin) – помните, что в пароле важен регистр!

Как бы мне?..

Если вы – начинающий пользователь Linux, то после установки CentOS у вас могут появиться вопросы о выполнении определенных задач. Вот список наиболее распространенных задач и способов их выполнения...

» **Работа в сети** Щелкните по оранжево-голубому значку на верхней панели, чтобы запустить Firefox, популярнейший в мире браузер с открытым кодом, отличающийся быстротой и надежностью.

» **Работа с файлами** Дважды щелкните по значку домашней директории на рабочем столе, чтобы получить доступ к своим документам.

» **Воспроизведение музыки и видео** Перейдите в Applications > Sound & Video, и вы найдете там подборку аудио- и видеопроигрывателей.

» **Работа с документами** В Applications > Office вы найдете OpenOffice.org, мощный пакет, совместимый с файлами MS Office.

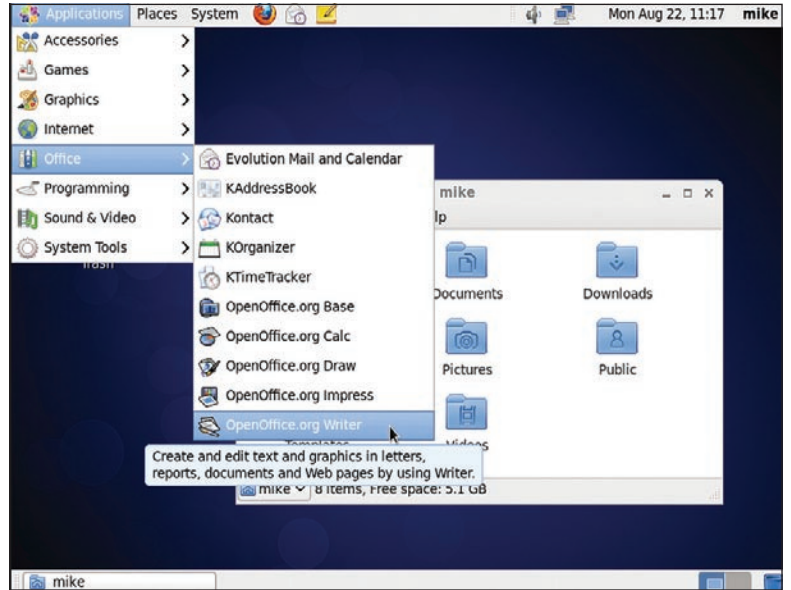
» **Редактирование изображений** Зайдите в Applications > Graphics и попробуйте GNU Image Manipulation Program, всесторонний инструмент в стиле Photoshop.

» **Настройка системы** В меню System вы найдете Preferences для настройки рабочего стола и Administration, с инструментами типа менеджер печати, настройка брандмауэра и т.д.

» **Поиск новых программ** Зайдите в System > Administration > Add/Remove Software, чтобы найти добавочные пакеты в Интернете.

» **Завершение работы** Щелкните по Menu > Logout, чтобы перейти в режим ожидания, перезагрузиться или выключить компьютер.

CentOS – дистрибутив, полностью поддерживаемый сообществом, и хотя он основан на исходнике RHEL, он существует совершенно отдельно от Red Hat, так что вы не можете просить Red Hat о поддержке. Но это не проблема: сайт CentOS на www.centos.org содержит ссылки на массу информационных ресурсов. И вы можете пользоваться обширной документацией Red Hat на <http://tinyurl.com/2w7dlbv>.



Если при работе с дистрибутивом возникнут проблемы, разместите сообщение на форумах на www.centos.org/modules/newbb/. Там вы увидите подфорумы, посвященные именно CentOS 6, с разделами по общим вопросам, поддержке оборудования и т.д.

Размещая вопрос, будьте предельно точны и цитируйте все выданные сообщения об ошибках; вам обязательно помогут! LXF

» CentOS работает как RHEL, но без ее логотипов и торговых марок.

Не пропустите...

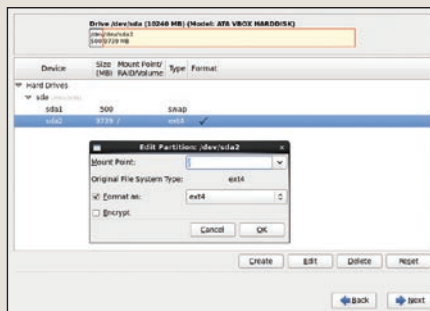
Gnome 2!
Если вы упорно не ладите с Gnome 3 или Unity, вам понравится этот классический рабочий стол.

Мега-обновления
Если все будет хорошо, то сообщество CentOS будет выпускать обновления до ноября 2017 года. Круто!



4 Разделы диска

CentOS спросит вас, хотите ли вы установить дистрибутив вместе с другой ОС или используете весь диск целиком. Для разбиения на разделы вручную выберите Custom Layout.



5 Ручное разбиение

Если вы выбрали эту опцию, создайте раздел root (/) в формате ext4 размером минимум 10 ГБ, и раздел подкачки [swp] размером вдвое больше ОЗУ, максимум 2 ГБ.



6 Выбор

Начнется копирование файлов, и по завершении процесса установки можно перезагрузиться, извлечь DVD и запустить вашу новую систему CentOS. Наслаждайтесь!

Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам. Но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в «Линуксцентр» по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:

 <p>LXF147 Август 2011</p> <ul style="list-style-type: none">» Fedora 15 Снимаем (красную?) шляпу перед героями разработки» Подкасты А не поговорить ли за жизнь» Моего сердца чемпион Матч на первенство дистрибутивов» Приключения GPL в России О поправках к законодательству <p>LXFDVD: Аж 9 дистрибутивов, LibreOffice 3.3.2, Asterisk 1.8 и многое другое...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_147/ PDF-версия: shop.linuxformat.ru/elxf_147/</p>	 <p>LXF148 Сентябрь 2011</p> <ul style="list-style-type: none">» Linux Mint 11 Он расшатал трон Ubuntu» Резервные копии онлайн Подальше положишь – поближе возьмешь...» Топ-50 Лучшие приложения, по мнению пользователей» Сажаем Eucalyptus Что нам стоит облако построить <p>LXFDVD: Все 50 лучших, Linux Mint 11, Mageia 1, Clonezilla 1.2.8, GIMP 2.6.11, Blender 2.58 и многое другое...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_148/ PDF-версия: shop.linuxformat.ru/elxf_148/</p>	 <p>LXF149 Октябрь 2011</p> <ul style="list-style-type: none">» Sabayon Gentoo с человеческим лицом» Читалки новостей Отводим от потока информации личный ручеек» GPL по понятиям Российские законы – о свободном ПО» И среди звезд, и в облаках Linux взлестусь» ПО, звони мне Тандем смартфона и компьютера <p>LXFDVD: Sabayon, PCLinuxOS, Fotoxx 11.7, Thunderbird 5, Armadillo 2.1.91, pyKoko 0.3.3 и многое другое...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_149/ PDF-версия: shop.linuxformat.ru/elxf_149/</p>
---	--	---

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала – оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru или shop.linuxformat.ru получают электронную версию в подарок!

Подписывайтесь на журнал на www.linuxformat.ru/subscribe/

Телефоны отдела подписки: Санкт-Петербург (812) 309-06-86, Москва (499) 271-49-54

Специальное предложение

Купите подборку журнала!

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски LXF. Если вы тоже этим озадачены, то в интернет-магазине «ГНУ/Линуксцентра» продолжается продажа журналов за 2010 год. Вы можете приобрести как отдельные номера изданий, так и подписки на 6 или 12 месяцев. Спешите – журналов осталось не так уж много!

shop.linuxformat.ru



6 месяцев
900 руб.

12 месяцев
1800 руб.

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
- » **имя_программы-1.0.1.tgz** – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

А ТАКЖЕ: LibreOffice 3.4.2, Firefox 5.0.1, Systemd и другие обновления

» Ядро 2.6.38.7 » KDE 4.6.5 » Mandriva Smart Desktop » MandrivaSync


Мандрива
Mandriva 2011. 32-разрядная сборка

Ноябрь 2011
LXF DVD 150

ЛINUX
FORMAT

Ноябрь 2011
LXF DVD 150

ЛINUX
FORMAT

Эталон надежности в мире Linux



CentOS 6

А ТАКЖЕ: Tty Core Linux 3.8.1, MediaWiki 1.7, Susegu 1.3.7 и многое другое...

ПЛЮС:



ЛINUX
FORMAT

Содержание

LINUX FORMAT

ДИСТРИБУТИВЫ

CentOS 6 Дистрибутив на базе Red Hat Enterprise Linux, поддерживаемый сообществом (встряхнул **LXF DVD**)
Chakra Project 2011.09 LiveCD со свежей версией KDE 4.7 (встряхнул с **LXF DVD**)
Mandriva 2011 Стартовый DVD, 32-разрядная сборка (встряхнул с **LXF DVD**)
Ubuntu 11.10 alpha 3 Предварительная версия осязаемого релиза (ISO-образ)

РАБОЧИЙ СТОЛ

DigiKam 2.0.0 Кросс-платформенная программа для управления коллекциями фото-графий
Dln 1.7 Синтезатор звуковых волн прокладочной формы
Evolve 04.9 Эмулятор терминала с возможностью смены кодировки в ходе сеанса
Gnome Subtitles 1.2 Редактор субтитров с возможностью перевода текста и просмотра видео

KeepPass 2.16 Менеджер паролей, сохраняющий их в зашифрованном файле на диске
PdfMaster 0.5.0 Программа конвертирования PDF в форматы MOBI и EPUB
Rubrica 2.0.10 Адресная книга, написанная на GTK+, с возможностью экспорта в HTML
Transite 0.11.8 Текстовый редактор с возможностями компьютерного перевода CAT

Venus 1.18 Программа построения графиков с возможностью вывода в Postscript, PDF и SVG

РАЗРАБОТКА

Dociffler 2.7 Средство перекодировки документации формата (n,l,gr) в DocBook XML
Fossil 1.18 Распределенная система контроля версий, включающая wiki и отслеживание ошибок
Google Go Свободная реализация языка программирования Go и учебный курс Р. Пайка

Object 2.0.3 Объектно-ориентированный язык программирования (компилятор и VM)
Piling 2.4.7 (Piling Is Not GNU Make) Система сборки на базе Apache Ant

ИГРЫ

BuggerSpace 1.9.0 Головоломка, сборка «лабиринта» из интринсенов
LGeneral 1.2.2 Позиговая стратегия, свободная реализация Panzer General
Teeworlds 0.6.1 Многопользовательская онлайн-игра, «пересечение» Quake и Worms
Tyran 0.334 Кросс-платформенная Roguelike-подобная игра, написанная на Java

НОТРИКСЫ

Anatark 2.4.3 Кросс-платформенный музыкальный проигрыватель
Archiver 3.2 beta Средство мониторинга и управления виртуальными машинами KVM
CKNet 0.1.6 Реализация игры KNet на cutils
Subosphere 0.1 Свободная реализация в Linux игры "Kula World 7" Roi Amau для gPadder 2.18
 Простая программа для загрузки и воспроизведения подкастов

IRyphon 0.11 Среда интерактивного программирования на Python (оболочка, средства визуализации, построения GUI и т.д.)
MiniTube 1.5 Приложение для просмотра роликов с YouTube
Rowscope 1.2 Программа для просмотра больших (более 1 Гб) текстовых файлов

Takeoff 0.2 Полноэкранное меню для KDE (Plasma) в стиле Launchpad на MacOS
Tiny Core 3.8.1 Максимально облегченный (10 Мб) дистрибутив, использующий Busybox, Tiny X и Fik (ISO-образ)
Usermin 1.4.80 Пользовательский web-интерфейс для работы с почтой на сервере IMAP

INTERNET

Bitlita 1.38 Клиент BitTorrent, выполняемый как демон
Data Mail 4.6.1 Менеджер списков рассылок с web-интерфейсом
Firefox 6.0 Обновленная версия популярного браузера

MediaWiki 1.17 Наиболее популярный сервер wiki, изначально разработанный для Wikipedia
Tin 2.0.0 Программа чтения новостей, работающая в текстовом режиме, с богатым набором функций фильтрации и обработки сообщений

СИСТЕМА

Firestarter 1.0.3 Межсетевой экран с удобным графическим интерфейсом для его настройки
Pungi 2.5 Средство сборки релизов из репозитория Fedora
renamutils 0.11.0 Набор утилит для переименования файлов
Synnergy 13.7 Программа, позволяющая управлять несколькими компьютерами под управлением Linux, Mac OS и Windows с помощью одной клавиатуры и мыши
UCK 2.4.4 Средство индивидуальной настройки официального Ubuntu Live CDS (Kubuntu/Xubuntu и Edubuntu)

Пожалуйста, перед использованием Аджнито Диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@lxf.com
ДЕЯТЕЛЬНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@lxf.com

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать апликативный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте и установите с его сайтом следующие программы:
 *Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАР 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor=0,3,0 12 16M
```

Первое слово в этой строке – метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду `cdrecord -v /path/to/image.iso`

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.



- ТЕЛЕФОНЫ
- НОУТБУКИ
- КОМПЬЮТЕРЫ
- ЭЛЕКТРОННЫЕ КНИГИ
- ФОТО И ВИДЕО
- КАРТЫ ПАМЯТИ
- АУДИО-ВИДЕО ТЕХНИКА
- ДОМАШНИЕ КИНОТЕАТРЫ
- GPS-НАВИГАТОРЫ

**ПОКУПКИ
И НЕМНОГО БОЛЬШЕ**

Школа LXF

Спонсор рубрики
Mandriva.ru
разработчик
дистрибутива
EduMandriva
www.mandriva.ru

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Школьные «Ответы»

Вопросы задают не только ученики, но и учителя – а Александр Казанцев на них отвечает.



Наш
эксперт

**Александр
Казанцев**

Старший научный сотрудник ЗАО «Тырнет» и автор образовательного проекта EduMandriva в настоящий момент посвящает все свое время скрепянию железа и софта под свободными лицензиями.

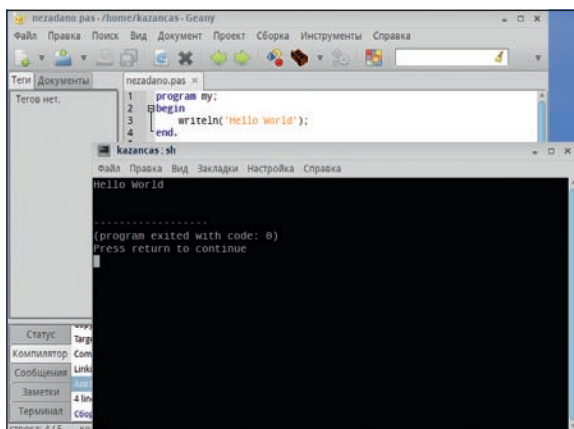
Учителя, которые используют Linux, задают множество вопросов, ответы на которые не занимают много места – и по отдельности из них не получится полноценное руководство или статья. Поэтому мы решили в этом номере собрать наиболее часто задаваемые вопросы и создать нечто похожее на Ответы LXF. Если такой формат вам понравится, то в дальнейшем мы можем сделать его постоянным, отведя одну полосу на просвещение преподавателей. Присылайте ваши вопросы на info@linuxformat.ru, чтобы у нас был повод продолжить рубрику.

1 Geany и консоль

В мне и ученикам очень нравится редактор Geany – за его минимализм и возможность работать со множеством языков программирования, а также функции автодополнения и другие полезные вещи. Но я работаю в KDE4, и программы в Geany запускаются в терминале XTerm, который вместо русских букв выводит кракозябры. Когда я ставлю терминал Konsole, Geany ничего туда не выводит. Что делать в таком случае?

О Да, Geany прекрасная программа, и она может работать с любым терминалом, но вот работа с Konsole вызывает проблемы не только у вас. Есть несколько путей решения. Первый заключается в использовании специфического вызова Konsole. Для этого необходимо в меню Правка > Настройки > Инструменты Geany в поле Терминал вписать следующие строки

```
konsole -e sh -c
```



➤ Вот теперь Hello World отображается корректно и в Konsole.

Второй способ заключается в замене xterm на lxterminal, который гарантированно есть в репозитории вашего дистрибутива. Установите его и впишите вызов в поле Терминал.

После этого можно будет продолжать работать с Geany и в KDE4.

2 Почему не компилирует?

В При попытке компиляции приложения в Geany или Lazarus выводится ошибка, что не найдена программа или файл. Из-за чего это может быть?

О Обычно такая ситуация возникает в том случае, если не найден компилятор или в пути к файлу с исходными кодами есть не латинские символы или пробелы. В первом случае убедитесь, что у вас установлено требуемое программное обеспечение. Geany сам по себе не отслеживает его наличие, и может сложиться мнение, что он автоматом поддерживает все, что прописано в меню. Для Паскаля нужен компилятор FreePascal (обычно это пакет fpc), а для Lazarus также нужен пакет с исходными кодами (скорее всего это будет fpc-src). Для Basic рекомендуем использовать компилятор FreeBasic, который можно настроить на совместимость с QBasic.

Если же компилятор установлен, следует проверить путь до исходных кодов программы. Во-первых, не называйте файл с программой русскими буквами. Во-вторых, сохраняйте программу в директории с латинским названием, а не как любят у нас в школе – на Рабочем столе или в директории Документы. В-третьих, избегайте пробелов в именах, заменяя их нижним подчеркиванием. То есть файл ento_moya_pervaya_programma.pas нужно называть ento_moya_pervaya_programma.pas.

Ну и, наконец, убедитесь, что программа сохранена перед компиляцией.

3 Где взять редактор

В По программе нам надо проходить текстовый редактор Блокнот и графический редактор Paint. А что делать в Linux?

О В Linux вы можете делать тоже самое. Главное – знать, что использовать. К примеру, на сайте <http://metodist.lbz.ru/authors/informatika/3/> вы можете найти компьютерные практикумы для Linux (подойдут не только для АЛБТ), рассчитанные на использование двух приложений: текстового редактора

В первом варианте обычно предлагается переназначить права на директорию и файлы конфигурации домашнего каталога на суперпользователя-root, командой

```
chown -hR root:root /home/uchenik
```

После этого достаточно будет выйти из текущей сессии и зайти заново, чтобы убедиться, что в системе ничего нельзя сломать. Но... без ложки дегтя тут не обойдется. Многие программы требуют возможности записи в домашнюю директорию, чтобы нормально функционировать: одни программы сохраняют временные файлы, другие используют локальные базы данных, третьи нуждаются в механизмах сохранения сессий. К примеру, в браузере перестанут работать службы Google вроде Gmail, перестанет запускаться LibreOffice и другое ПО.

Как вариант, можно использовать возможности типа пользователя xguest в Mandriva, когда данные сессии не сохраняются, или разворачивать текущую сессию во временной директории или памяти; но эти решения неординарны по сути и, к сожалению, у большинства преподавателей вызовут массу проблем.

Поэтому мы пойдем по второму пути и научимся делать резервную копию домашней директории и быстро ее восстанавливать на прежнее место.

Самый простой способ – сразу же после установки настроить все нужные программы и создать архив (сразу предупреждаем, что это не сжатый архив типа zip или rar) с резервной копией. Причем делать это лучше не при запущенной сессии пользователя. Для этого выйдите из архивируемого пользователя и переключи-

тесь в терминал (с помощью Ctrl+Alt+F1–F6), затем войдите в систему с административными правами (от имени root) и завершите графический режим

```
init 3
```

После этого создайте нужный нам архив для последующего восстановления:

```
tar -Puf backup.tar /home/user
```

Далее, когда детишки в очередной раз что-нибудь сломают, можно будет или дать команду (от имени root):

```
tar -xvf backup.tar -C /
```

или просто открыть архив в mc и скопировать все файлы назад. Для надежности вернем права пользователю:

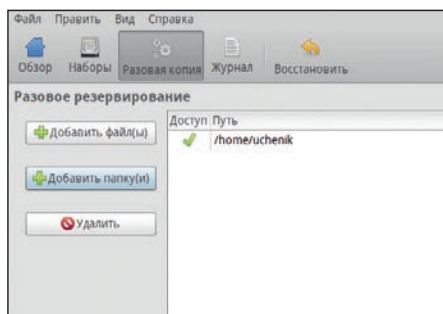
```
chown uchenik:uchenik -hR /home/uchenik
```

Если вы не очень ладите с командной строкой, можете воспользоваться программами для создания резервных копий. Программы типа *DejaDup*, *BackInTime*, *Fwbackups* помогут вам получить архив с копией нужного домашнего каталога пользователя, восстановить все из копии, выбрать для архивирования только требуемые файлы (к примеру, не стоит архивировать каталоги без точки в начале, так как они будут содержать данные учеников, документы, тексты программ, а не настройки). Кроме того, этим программам можно поручить выполнять резервное копирование в инкрементном режиме по определенному графику, создавая подобие точек восстановления Windows. И все это потребует от вас только щелчков мыши (ну, может быть, еще ввода названия соответствующего архива).



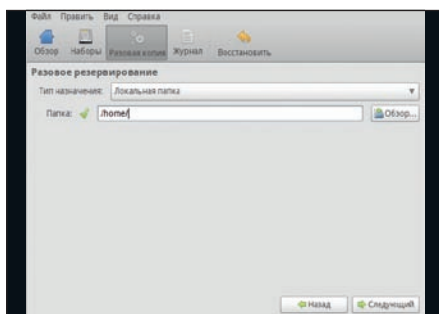
Шаг за шагом: Работаем с резервными копиями

Создадим и восстановим резервную копию директории с помощью fwbackups.



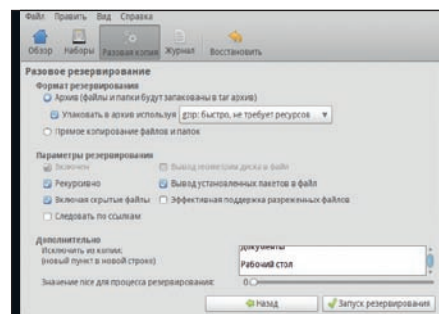
1 Приступим

Выбираем разовое резервирование и добавляем директорию, содержимое которой мы собрались сохранять.



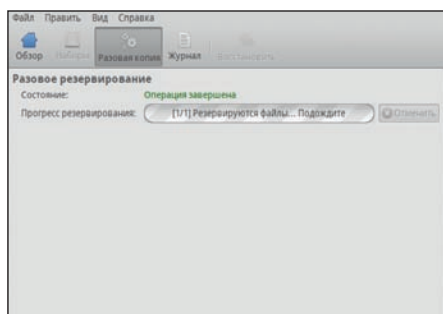
2 Назначим архив

Нажав Следующий, выбираем место хранения и тип создаваемого архива.



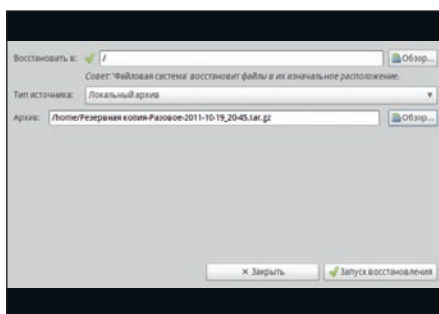
3 Отберем данные

Выбираем тип сжатия, а также исключаем из копии **Документы, Загрузки, Рабочий стол, Изображения, Видео, Шаблоны и Музыка**.



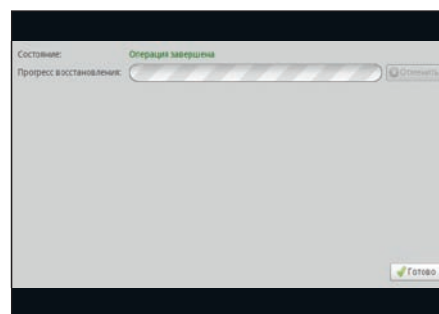
4 Создадим архив

Запускаем резервирование и получаем в /home архив вида **Резервная копия-Разовое-2011-10-19_20-45.tar.gz**.



5 Восстановим из архива

При необходимости восстановить данные выбираем **Восстановить** и указываем назначение в /, тип источника **Локальный архив** и ссылку на него.



6 Все готово

После нажатия **Запуск восстановления** мы возвращаем исходные данные на их место.

5 Глаза разбегаются

ВЯ окончательно запуталась в этих средах рабочего стола, терминалах, эмуляторах, редакторах и прочих премудростях Linux. Мне нужно более понятно рассказать, что нужно поставить на машины с 256, 512 и с 1 и более гигабайтами памяти и где что искать. Я хоть и информатик, но не системщик и такими премудростями не обладаю.

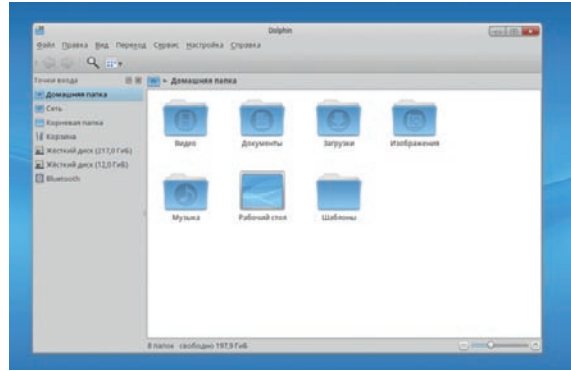
ОНичего страшного. Мы иногда сами путаемся в версиях программ и их названиях, благо мир OpenSource предполагает множественные решение одной задачи.

Вы правильно решили опираться на объем оперативной памяти – в настоящий момент именно эта характеристика влияет на выбор учебного дистрибутива.

Если у вас в компьютере 128–256 МБ ОЗУ, то выбор будет обращен в сторону легковесных дистрибутивов со средами рабочего стола (их еще называют окружениями, рабочим столом, Desktop Environment или просто DE) *LXDE* или *Xfce*. Они неприхотливы, имеют интерфейс, похожий на Win98/XP, и нетребовательны к ресурсам. В них вы сможете найти терминалы (*Lxterminal* и *Terminal* соответственно), файловые менеджеры (*PCManFM* и *Thunar*), текстовые редакторы (*Leafpad* и *Mousepad*).

Если у вас более 256 МБ, но менее 512, то вам могут подойти DE Gnome версии 2 или KDE 3. Также вы сможете установить KDE 4, но тут желательно иметь видеокарту с 3D-ускорением и поддержкой в лице проприетарных драйверов.

Если же у вас 1 ГБ ОЗУ и более + современная видеокарта (даже встроенная в процессор от Intel или AMD), то смело ставьте последние версии DE – Unity, KDE4 или Gnome3.



➤ Научившись определять DE, вы сразу сможете сказать, что это KDE 4, и в нем можно использовать то-то и то-то.

В Gnome и Unity вы найдете файловый менеджер *Nautilus*, терминал – *Gnome-terminal* и текстовый редактор *Gedit*. В KDE 4 соответственно будут *Dolphin*, *Konsole* и *KWrite*.

Эмуляторы в Linux бывают разных видов – «не эмуляторы» вида *Wine*, который является слоем совместимости и позволяет запускать (не все) приложения Windows; эмуляторы ОС типа *DosBox* или *Dosemu*, позволяющие запускать старые программы DOS, типа *Кенгуренок* или *TurboProlog*; а также эмуляторы полноценных систем типа *VirtualBox* или *QEMU*, которые дают возможность установить в Linux другую операционную систему – Windows, MacOS или вообще какую-нибудь экзотическую.

Чтобы понять, какие программы вам нужны для использования по аналогии с Windows, советуем прочитать статью из майского **LXF144** «В едином стиле». Надеемся, это поможет вам разобраться в программном обеспечении. **LXF**



Академическая программа для учебных заведений

Mandriva Linux

Mandriva.Ru предоставляет учебным заведениям лицензию, дающую право на неограниченное по числу рабочих станций использование дистрибутива Mandriva Linux на всех компьютерах в образовательном учреждении, всех компьютерах преподавателей и всех компьютерах учащихся, в том числе и домашних.

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



В декабрьском номере

Покоритесь, команды!

Bash вам будет не страшен – из зеленых новичков мы делаем матерых хакеров экстра-класса.

Возрождение железа

Зачем же выбрасывать эти бежевые башни?
Вернем их на ваш стол и найдем им применение.

Строим медиа-центр XBMC

Научимся создавать и эксплуатировать крутые медиа-центры, не мучаясь с *MythTV*.

Содержание последующих выпусков может меняться без уведомления: Грэма ждет декретный отпуск.

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор
Кирилл Степанов info@linuxformat.ru

Литературный редактор
Елена Толстякова

Переводчики

Илья Аввакумов, Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова, Татьяна Цыганова

Редактор диска

Кирилл Степанов

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатели

Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Скай ЛТД»

196210, Санкт-Петербург, Вазетная ул., 11, корп. 2, лит. А

Тел.: (812) 677-98-03

Заказ 4136

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.com

Заместитель редактора Эндрю Грегори [Andrew Gregory]

agregory@futurenet.com

Редактор по продвижению и сообществу Майк Сондерс [Mike Saunders]

mike.saunders@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

Подготовка материалов Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown],

Алекс Кокс [Alex Cox], Бен Эверард [Ben Everard], Марко Фиоретти

[Marco Fioretti], Эндрю Грегори [Andrew Gregory], Мэтт Хэнсон [Matt Hanson],

Боб Мосс [Bob Moss], Адам Оксфорд [Adam Oxford], Джонатан Робертс

[Jonathan Roberts], Маянк Шарма [Mayank Sharma], Шашанк Шарма

[Shashank Sharma], Майк Сондерс [Mike Saunders], Ник Вейч [Nick Veitch],

Евгений Балдин, Александр Казанцев, Тимур Мубаракшин, Тихон Тарнавский,

Андрей Ушаков, Алексей Федорчук, Роман Ярыженко

Художественные ассистенты Стейси Блэк [Stacey Black],

Ник Кокс [Nick Cox], Фил Хейкрафт [Phil Haycraft]

Иллюстрации Крис Хедли [Chris Hedley], Крис Винн [Chris Winn],

Ely Walton Illustrations, iStockPhoto

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Tel. +44 01225 442244 Email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):

Лиговский пр., 50, корп. 15

Тел. +7 (812) 309-06-86

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС 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 ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

LINUX – зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

«GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futurepic.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

Защита для высокоскоростных сетей Ethernet 100/1000 Мбит/с

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

ВЫДЕЛЕННЫЕ СЕРВЕРЫ

HETZNER
ONLINE

МОЩЬ НА ПОЛНУЮ!

НОВОЕ ПОКОЛЕНИЕ СЕРВЕРОВ HETZNER

Новинка!



HETZNER DEDICATED ROOT SERVER EX 4

- Intel®Core™ i7-2600 Quad-core с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM
- 2 x 3 ТБ SATA 6 Гбит/с 7200 об/м! (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- Установка из образов
- Без минимального контракта
- установка 5 500 рублей

1900
рублей в месяц

HETZNER DEDICATED ROOT SERVER EX 6

- Intel®Xeon® E3-1245 Quad-core с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM ECC
- 2 x 3 ТБ SATA 6 Гбит/с 7200 об/м! (Software-RAID 1) Enterprise class
- Операционная система Linux
- Неограниченный трафик*
- Установка из образов
- Без минимального контракта
- установка 5 500 рублей

2900
рублей в месяц

HETZNER DEDICATED ROOT SERVER EX 8

- Intel®Xeon® E3-1275 Quad-core с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM ECC
- До четырех жестких дисков на выбор
- Операционная система Linux
- Неограниченный трафик*
- KVM-over-IP для постоянного доступа
- Установка из образов
- Без минимального контракта
- установка 5 500 рублей

3500
от 3500
рублей в месяц

*Трафик предоставляется бесплатно. При превышении порога 10000 ГБ/месяц скорость соединения будет ограничена до 10 MBit/s. Стоимость аренды постоянного канала с пропускной способностью 100 MBit/s составляет 290 рублей за каждый дополнительный ТБ.



Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выберите более чистое будущее вместе с Hetzner Online!

RU.HETZNER.COM
info@ru.hetzner.com