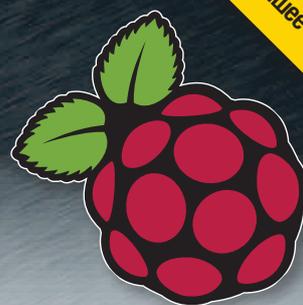


**ЦЕЛЫХ 15 ДИСТРИБУТИВОВ!**  
2 стороны диска, набитых всяким добром  
Плюс: лучшее новое ПО и более того!

# LINUX FORMAT

Главное в мире Linux



Raspberry Pi: Подключение отовсюду с. 66

Октябрь 2012 № 10 (162)

# ЛУЧШИЙ ДИСТРИБУТИВ-2012

Мы столкнули лбами ярчайшие релизы года — пусть победят за первенство на рабочем столе



**ПЛЮС!**  
Инсталляция с openSUSE LiveCD не лишена приятных сюрпризов с. 54



## Opera против Chrome

» Браузерные войны 2: толстые тетки поют, а они все воют с. 14



## Карты и выход из кризиса

« Открытым должно быть не только ПО, но и знание в целом »

Дэвид Кобиа: технарь-гуманист с. 42

## Также в номере...

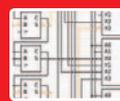
### Грызня онлайн

Отчего разгораются войнушки и зачем людям асбестовые штаны... с. 46



### Виртуально строим CPU

Разберемся, что творится внутри вашего ОЗУ с. 50



### Изучим Контейнеры Linux

Виртуализация по частям – контроль проще с. 88



Свободное ПО  
Лицензия мечты

» Выберите подходящую для своего проекта

Безопасность — легко  
Правильный пароль

» Девичья фамилия вашей матушки — еще не защита

Врага надо знать  
SecureBoot

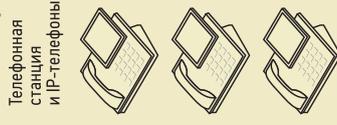
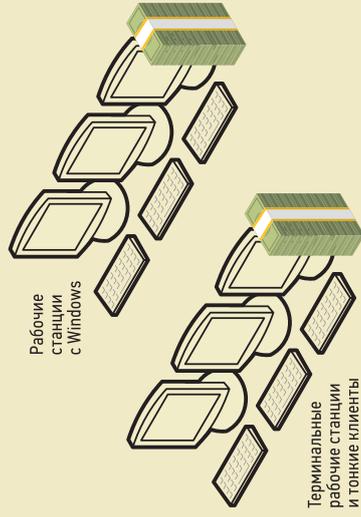
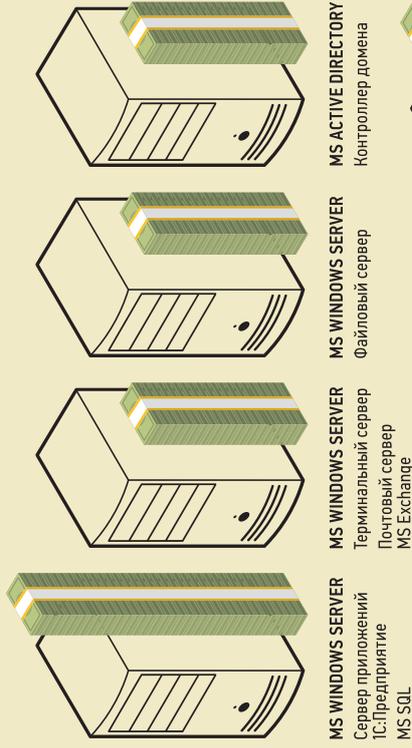
» Как Microsoft норовит поломать кайф от Linux

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

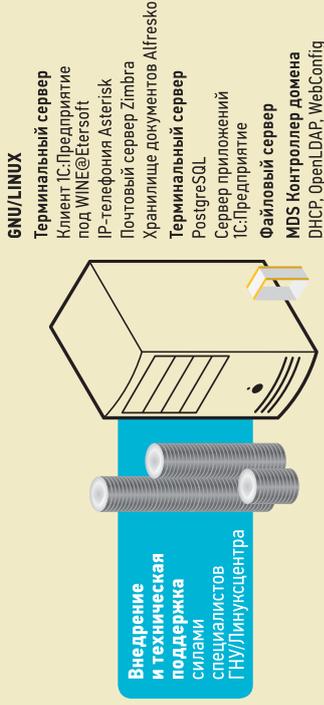
Linux center  
www.linuxcenter.ru

# ВИРТУАЛЬНАЯ ИТ-ИНФРАСТРУКТУРА ПРЕДПРИЯТИЯ НА БАЗЕ СВОБОДНОГО ПО

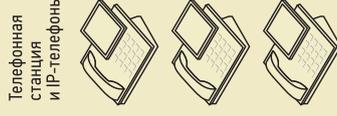
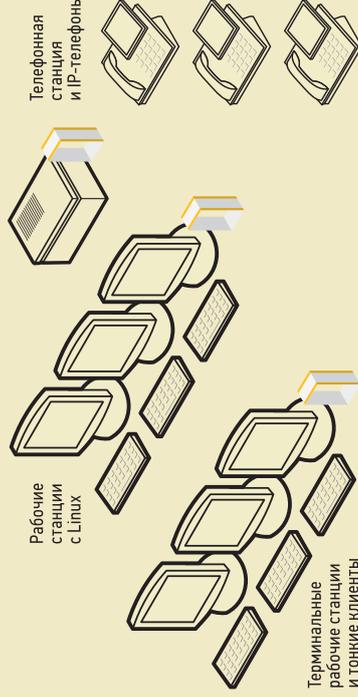
## На базе Windows\*



## На базе GNU/Linux и виртуализации\*



**Почтовый сервер, файловый сервер, контроллер домена, базы данных SQL, IC:Предприятие, IP-телефония – и все это на одном физическом сервере в виртуальных машинах!**



## ПРЕИМУЩЕСТВА ТЕРМИНАЛЬНОГО РЕШЕНИЯ НА БАЗЕ GNU/LINUX:

- экономия средств на лицензировании ПО
- сокращение затрат на оборудование за счет виртуализации
- единая инфраструктура на базе свободного ПО
- защита от вирусов и других угроз
- безопасности

**Позвоните сейчас!**  
Назначьте встречу с нашим специалистом!

**Офис в Санкт-Петербурге**  
Лиговский пр., 50, корпус 15  
Тел.: (812) 309-06-86,  
факс: (812) 640-49-90

**Офис в Москве**  
Красноказарменная ул., 17  
(в помещении АТС МЭИ)  
Тел.: (499) 271-49-54



\* Сравнительная стоимость программного обеспечения в комплексном решении для малого и среднего бизнеса

# Свободное ПО + виртуализация = экономия в 10 раз!

Лицензионные платежи отсутствуют, планируются бюджет только на внедрение и техническую поддержку

ВЫСТАВКА

КОНФЕРЕНЦИЯ

ПРЕМИЯ



# RIW//12

RUSSIAN INTERNET WEEK

**17/18/19** ОКТЯБРЯ

Москва, Экспоцентр на Красной Пресне

ОРГАНИЗАТОР

**РАЗ<sup>+</sup>К**

СТРАТЕГИЧЕСКИЙ ПАРТНЕР

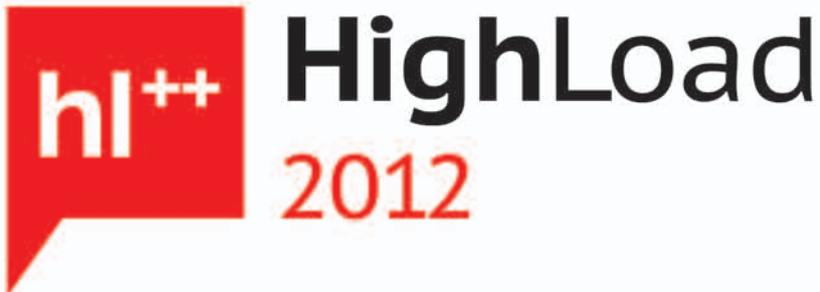


**Ростелеком**  
Больше возможностей



[WWW.RIW12.COM](http://WWW.RIW12.COM)

# Шестая профессиональная конференция разработчиков высоконагруженных систем



**В программе доклады  
лучших специалистов  
Рунета**

Архитектуры

Оптимизация баз данных

Системное администрирование

Операционные системы (Linux, Solaris, FreeBSD)

Организация разработки  
высоконагруженных проектов

Тестирование: функциональное и нагрузочное

Примеры работающих решений

[www.highload.ru](http://www.highload.ru)

## Что мы делаем

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

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

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

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



## Кто мы

Вопрос, вдохновленный статьей Майка про дебаты в Linux на стр. 46: что в Linux раздражает вас больше всего?



**Гэри Уокер**

Почему бы там не быть одному хорошему дистрибутиву, а не куче? Джон мне говорил, что они все равно одинаковые.



**Эндрю Грегори**

KDE против Gnome против Mate против Cinnamon против Unity. Может, лучше принимать в день по таблеточке?



**Эфраин Эрнандес-Мендоса**

Как Ричард Столлмен выживает на диете из этого его сыра? Я предпочитаю пирожки с начинкой...



**Бен Эверард**

Почему в Gnome Shell нет кнопки ВЫКЛ? Мне со сломанной рукой понадобилось шесть недель, чтобы его вырубить.



**Маянк Шарма**

Отчего Шашанк Шарма так фатально безнадежен в игре SuperTuxKart?



**Джонатан Робертс**

Отсутствие поддержки корнваллийского языка в Libre Office.



**Майк Сондерс**

Выбора там слишком много...



**Валентин Синицын**

Две гласных буквы: когда я слышу «линукс», я хватаюсь за пистолет. Fgaax было бы лучше.



**Ник Вейч**

Почему после стольких лет «они» по-прежнему упорно отказываются изменить название Linux на Veitch?



**Сюзан Линтон**

Логичные решения высокородных баронов выпуска LXF – что делать курсивом, а что жирным шрифтом.



**Шашанк Шарма**

Почему Fedora не делает вегетарианскую версию Мясистого Чуда? Титан Тофу, например?



**Нейл Ботвик**

Как подкаст LXF сочетает бесплатность с остроумием и глубиной? И в чем проблема женщин-энтов?



## Сто цветов

» Темой данного номера стало сравнение дистрибутивов, вышедших в первой половине 2012 года. На этот раз – «по олимпийской системе с выбыванием». В каждом раунде пара дистрибутивов сравнивалась по одному-единственному критерию, и выбирался победитель. В финал вышли... Впрочем, не буду пересказывать статью. А на обеих сторонах диска этого месяца целых 15 дистрибутивов. Так, для сравнения.

По моему глубокому убеждению, споры о «лучшем дистрибутиве всех времен и народов» – дело заведомо гиблое. После десяти минут обсуждения все сводится к сугубо субъективным оценкам «нравится – не нравится». Вероятно, при желании можно доказать теорему о том, что один дистрибутив может быть превращен в другой за некое конечное число шагов по установке и удалению пакетов и сборке чего-то недостающего из исходников. То есть технологические отличия отсутствуют, остаются только эстетические. И причина их появления – постоянное стремление к совершенству.

Тут уместно вспомнить фразу из канонического цитатника Мао Цзедуна «Пусть расцветают сто цветов, пусть соперничают сто школ» – лозунг, который китайский лидер провозгласил еще в 1957 году, задолго до появления свободного ПО.

**Кирилл Степанов**

Главный редактор

» [info@linuxformat.ru](mailto:info@linuxformat.ru)

## Как с нами связаться

Письма для публикации: [letters@linuxformat.ru](mailto:letters@linuxformat.ru)

Подписка и предыдущие номера: [subscribe@linuxformat.ru](mailto:subscribe@linuxformat.ru)

Техническая поддержка: [answers@linuxformat.ru](mailto:answers@linuxformat.ru)

Проблемы с дисками: [disks@linuxformat.ru](mailto:disks@linuxformat.ru)

Общие вопросы: [info@linuxformat.ru](mailto:info@linuxformat.ru)

Вопросы распространения: [sales@linuxformat.ru](mailto:sales@linuxformat.ru)

Web-сайт: [www.linuxformat.ru](http://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

Меч добрый, и тверда рука! И верные сердца... эх, Джонатан...

## Обзоры

### Opera против Chrome ..... 14

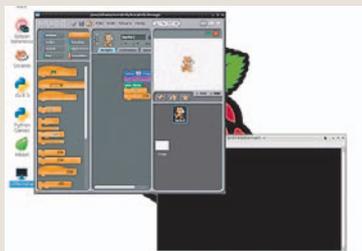
Мы столкнули лбами две тяжеловесных альтернативы *Firefox* в нашем тесте.



► Найдите отличие: *Opera* и *Chrome* в битве за главенство среди браузеров.

### Raspbian ..... 17

Raspberry Pi получил новый дистрибутив, и в его оценке мы разрывались между нашим внутренним технарем и настольным пользователем.



► Raspbian только что удостоен звания Рекомендованного дистрибутива для Pi.

### Alien Arena ..... 18

Заглушим тоску очередного каторжного дня в Future Publishing, малость поубивав пришельцев.



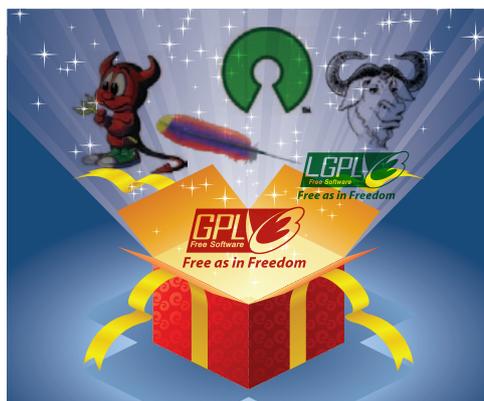
► Именно в таких сумрачных лабиринтах и следует гнездиться силам зла.

### Skype 4 ..... 19

Мы уж было решили, что, чего доброго, видели последний *Skype* на Linux, но – ура! Все не так, ребята!



Лучший дистрибутив 2012:  
Кто выйдет победителем  
в битве титанов за господство  
на настольном ПК? с. 34



Сравнение: Лицензии с. 28



Что за штука –  
SecureBoot? с. 58

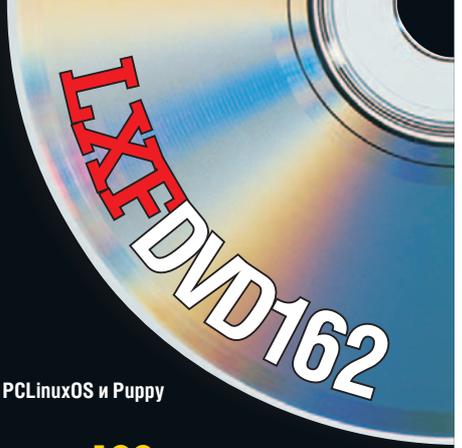
## Люди говорят



« Это до сих пор сносит мне мозг. Сроду не считал себя гуманистом. Прямо бред какой-то »

Дэвид Кобиа, технический руководитель Ushahidi с. 42

# На вашем бесплатном DVD



Октябрь 2012  
**LXF DVD 162**

**LINUX FORMAT**

**8** лучших дистрибутивов 2012 года

ЧЕМПИОНЫ ПО ВЕРСИИ LINUX FORMAT:  
» Mint » Mageia » openSUSE » CentOS

» Также: Arch, Fedora, PCLinuxOS и Puppy

## ЦЕЛАЯ ФЕЕРИЯ ДИСТРИБУТИВОВ

» 8 лучших Линуксов, которыми располагает мир свободного ПО

Диск включает: Arch, CentOS, Fedora, Mageia, Mint, OpenSUSE, PCLinuxOS и Puppy

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

### Ищите в этом номере



**Linux-дебаты** ..... **46**

На их фоне «Фрост против Никсона» – так, семечки.

**CPU своими руками, часть II** ..... **50**

Второй шаг на пути к рабочему процессору.

**Сисадминам** ..... **60**

Новые мудрые советы ученого Доктора.



## Пропустили номер?

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



### Учебники

**Хакерство Raspberry Pi** ..... **66**

Примените свой Pi, чтобы восторжествовать над шпионами и цензорами посредством SSH-туннелирования.

**Безопасность Пароли** ..... **70**

Не пора ли вам пересмотреть свой идентификатор пользователя и стратегию выбора пароля?

**Резервирование Облачное хранилище** ..... **74**

Готовить пищу в нетрезвом виде глупо: от этого бывают пожары в доме, приводящие к потере данных.

**Обработка текста Документация** ..... **76**

Создание первоклассной документации поможет пользователям щелкать ваши проекты как орехи.

**Электроника Arduino** ..... **80**

Не пугайтесь диаграмм: взбодрите свое устройство ввода матрицей клавиатуры.

**Безопасность Оболочки NMap** ..... **84**

Сетевым сканером командной строки можно управлять и через графический интерфейс.

**Виртуализация Контейнеры Linux** ..... **88**

Добавьте к своему парку машин еще одну, практически не нагружая ресурсы.

### Постоянные рубрики

**Новости** ..... **4**

Планируется портал на базе СПО для обсуждения законов, вышел проект «умного дома», а разработчики озвучены судьбой настольного Linux. И отпразднован День свободы ПО!

**Android** ..... **22**

Гуру от фотографии Джульетта Кемп научит вас размещать изображения на карте, грея душу удовлетворением от процесса. Мобильность позволяет это делать хоть с велосипеда.

**Сравнение** ..... **28**

Бен Эверард идет своим путем и выставляет оценки лицензиям, которые защищают вашу свободу и сохраняют открытость программного обеспечения.

**Интервью LXF** ..... **42**

Если бы не Дэвид Кобиа, гаитянам пришлось бы туго: последствия землетрясения они вряд ли преодолели бы в таком темпе.

**Что за штука** ..... **58**

SecureBoot: объясняем вмешательство Microsoft в ваш процесс загрузки.

**Рубрика сисадмина** ..... **60**

Более подробный путеводитель по тернистой тропе к администратору-звезде.

**Ответы** ..... **92**

**ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!**

Нейл Ботвик предоставляет помощь по оборудованию, беспроводной связи и дистрибутивам.

**Hotpicks** ..... **96**

Отдайте горяченького: лучшие в мире новинки свободного ПО.

**Диск Linux Format** ..... **102**

Содержимое двустороннего DVD этого месяца.

**Пропустили номер?** ... **104**

Оживите золотую эпоху, когда Джон с Майком оба еще работали на нас...

**Школа LXF** ..... **108**

Тяжело детство кактусов в условиях сугубо континентального северного климата. Поможем им!

**Через месяц** ..... **112**

Ретро-Linux, *Esoteric Bash* и *OpenStreetMap* ждут отважных исследователей.



**ГЛАВНОЕ** Государственный заказ » «Умный дом» на Java » Полное совершеннолетие  
» Платить или не платить за СПО? » Праздник свободы ПО

## ПЛОЩАДКА ДЛЯ ДИСКУССИЙ

# Будем судить по закону

«ПингВин Софтвер» разработает портал для обсуждения законопроектов.



» Рубрику готовил  
**АРТЕМ ЗОРИН**

В середине августа российская компания «ПингВин Софтвер» выиграла конкурс Министерства экономического развития по созданию интернет-портала для публичного обсуждения проектов и действующих нормативных актов органов власти субъектов России. Ориентировочно, проект будет реализован до ноября 2012 года.

Портал [regulation.gov.ru](http://regulation.gov.ru) является одним из инструментов реализации государственной программы «Информационное общество», рассчитанной на 2011–2020 годы, согласно которой необходимо привлекать широкие слои граждан к обсуждению проектов нормативно-правовых актов. Мероприятия в данной области предназначены для повышения прозрачности деятельности государства, а также проведения экспертной оценки и выявления положений, препятствующих осуществлению предпринимательской и инвестиционной деятельности.

[Regulation.gov.ru](http://Regulation.gov.ru) – ресурс федерального уровня, служащий для размещения фе-

деральными органами исполнительной власти проектов нормативно-правовых актов и результатов их публичного обсуждения. Задачей «ПингВин Софтвер» является модернизация существующего портала и создание на его основе типового решения для разворачивания аналогичных площадок в регионах на более низких уровнях властных структур.

## «Необходимо привлечь граждан к обсуждению.»

В рамках проекта специалисты «ПингВин Софтвер» разработают техническое задание на создание типового портала, включая схемы развертывания в регионах, создадут программное решение, позволяющее гибко управлять всеми порталами и масштабировать функционал, создадут шаблон внешнего вида типового портала, а также разработают механизмы консолидации данных, сквозного по-

иска по региональным порталам и сбора статистики для публикации на федеральном портале.

Для «ПингВин Софтвер» создание портала для обсуждения нормативно-правовых актов – не первый проект подобного рода. В 2009 году компания реализовала масштабный тематический интернет-портал [www.spohelp.ru](http://www.spohelp.ru), посвященный поддержке свободного ПО. В начале сентября 2012 выйдет в промышленную эксплуатацию онлайн-ресурс для контроля исполнения Федерального закона № 210-ФЗ «О предоставлении государственных и муниципальных услуг», который реализуется при участии комиссии Совета Федерации по развитию информационного общества. Другими словами, в конкурсе победил лостойный.



## А что же с НПП?

Напомним, что «ПингВин Софтвер» в конце 2011 года победила в конкурсе по разработке дистрибутива на основе Linux, который должен был стать основой Национальной программной платформы. Дело было сделано за рекордно короткое время – чуть менее месяца понадобилось специалистам компании, чтобы разработать прототип так называемой «национальной ОС». Одному Богу известно, сколько людей поседело, выполняя свой долг вовремя...

Однако с приходом нового министра связи Николая Никифорова в СМИ поползли слухи о том, что проект НПП закроют, а вместо создания сложного комплекса программ будет заключено соглашение с компанией Microsoft. Предполагается, что корпо-

рация установит для государства специальные цены, что позволит повысить эффективность закупок ПО.

Как сообщалось, новый министр считает, что развитие национальной программной платформы бесперспективно. Отмечается, что на Петербургском экономическом форуме Николай Никифоров уже встречался с руководством Microsoft: речь шла о потенциальном сотрудничестве, в частности, о применении «облачных» технологий в работе органов государственной власти и бизнеса. Вместе с тем коммерческие контракты пока не обсуждались.

Реакция министра связи последовала в его Твиттере, где он ответил на вопрос одного из читателей касательно судьбы НПП:

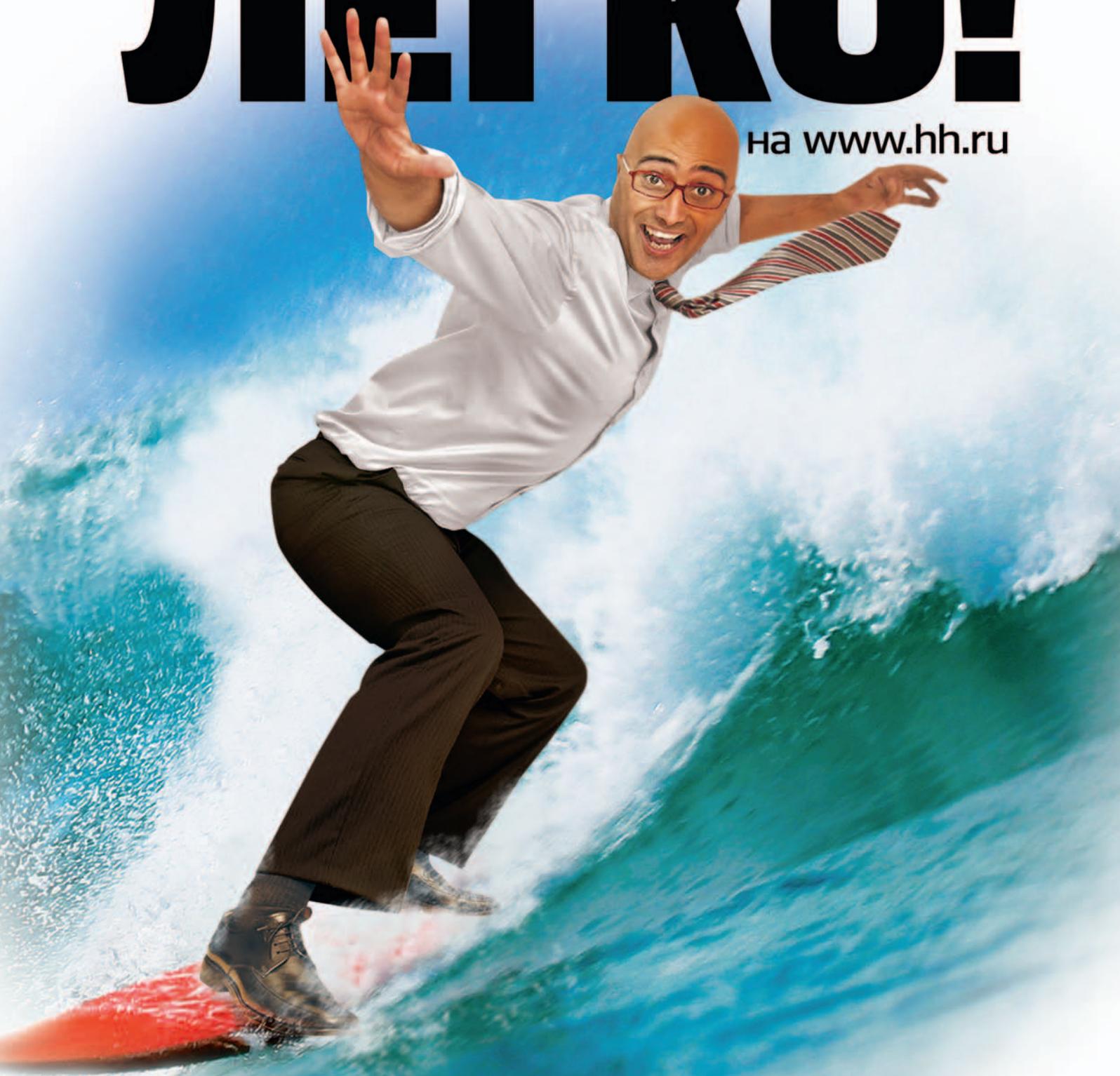
«Я такого не говорил, с чего Вы взяли?.. Не стоит так буквально доверять всему, что написали в газете». Но далее Николай Никифоров дополнил свой ответ: «...но нужно очень четко понять, чего же мы все хотим от НПП, на что и зачем мы тратим деньги налогоплательщиков».

В то же время, отдельные участники рынка считают, что отказ от развития НПП негативно отразится на информационной безопасности России.

В данный момент вопрос с НПП повис в воздухе. Пока известно лишь одно – новый министр намерен разобраться в целесообразности создания Национальной программной платформы, жизненно необходимой нашей стране.

# НАЙДИ РАБОТУ ЛЕГКО!

на [www.hh.ru](http://www.hh.ru)



**hh** **ru**  
HeadHunter

Выбирай из более чем 200 000 вакансий

НАСТОЛЬНЫЕ СИСТЕМЫ

# Слишком быстрые перемены?

В год 21-летия Linux ключевые разработчики обратили внимание на основные трудности.

**В** августе 2012 года ядро Linux отметило свой 21 день рождения: 25 августа 1991 года Линус Торвалдс объявил в телеконференции `comp.os.minix` о создании рабочего прототипа новой ОС Linux. По данным исследования, проведенного по заказу Евросоюза, приблизительная стоимость разработки с нуля проекта, аналогичного современному ядру Linux, составила бы более миллиарда долларов.

В связи с юбилеем некоторые ключевые разработчики экосистемы Linux обозначили основные проблемы Linux на «десктопе».

Например, издание *Wired* упомянуло мнение Мигеля де Икаса [Miguel De Icaza], инициатора проектов *Gnome*, *Midnight Commander* и *Mono*, в статье, задавшей вопрос «почему Mac OS X более успешна, чем Linux». Он сказал, что Linux отстает на рынке настольных систем, потому что разработчики инструментария для создания графических приложений для Linux должным образом не заботятся о сохранении обратной совместимости API. Из-за этого через некоторое время код перестает работать в новых дистрибутивах.

По части обратной совместимости Mac OS X много лучше, причем предоставляет Unix-подобное окружение; и разработчики приложений все чаще склоняются к данной платформе. Более того, со смещением серверных разработок в сторону Web, используемая на рабочей станции система перестает влиять на процесс разработки, и Mac OS X применяется для вы-

полнения работ, впоследствии используются на Linux-серверах.

Мигель де Икаса опубликовал в своем блоге более развернутые размышления о проблемах, мешающих развитию десктопа на базе Linux. Первой названа чрезмерная стремительность перемен; программа для одной версии дистрибутива может не заработать в следующем его выпуске. Особенность сообщества энтузиастов – малый интерес к рутине: основные силы тратятся на инновации, даже если они приводят к нарушению обратной совместимости. Вторая причина – несовместимость дистрибутивов (ключевых компонентов и версий библиотек); это не дает сформироваться единой экосистеме разработки, и силы распыляются.

Другие разработчики Linux, включая Линуса Торвалдса и Алана Кокса [Alan Cox], попытались это опровергнуть.

Линус Торвалдс подчеркнул, что одно из главных правил разработки ядра – сохранение внешних программных интерфейсов. Да, внутренние интерфейсы ядра находятся в постоянной переработке и часто меняются, но изменения остаются под капотом и не влияют на работу пользовательского окружения.

По мнению Линуса, отсутствие ограничений на внутренние изменения – один из важных факторов поддержания интереса к разработке ядра Linux и его популярности в среде программистов. Ядро пытается учесть интересы различных категорий пользователей, от систем с тыся-

» Даже наиболее дружелюбные к пользователю дистрибутивы Linux не решают всех проблем с популярностью ОС.



чами процессоров до встраиваемых платформ, телефонов и маршрутизаторов. А в *Gnome* разработчики придерживаются правила «мы лучше знаем, что им надо» и внедряют новые механизмы, нравятся они пользователям или нет, воспринимая все возражения как помеху прогрессу.

Мигель де Икаса ответил, что успех настольных систем связан не только с приложениями, но и с поддержкой оборудования системой. Непостоянство ABI-интерфейса усложняет разработку драйверов и требует их постоянной адаптации и пересборки для новых версий ядра. Из-за отсутствия двоичной совместимости, драйвер (и любой другой модуль ядра), собранный для одной версии ядра, не сможет работать в других версиях ядра без пересборки. Постоянные изменения в API подсистем печати, звуковой системе, шинах обмена данными и системе инициализации, вкупе с большой фрагментацией дистрибутивов, затрудняют работу сторонних коммерческих поставщиков приложений, для которых поддержка настольных решений под Linux обходится слишком дорого.

Алан Кокс встал на сторону Торвалдса, указав, что исполняемый файл с игрой, собранной им в 1992 году, до сих пор прекрасно работает независимо от версии ядра. Доводы об изменении звукового API ошибочны: прослойка совместимости с OSS допускает использование даже самых ранних звуковых приложений. Более высокоуровневые системы типа *PulseAudio* тоже имеют средства обеспечения совместимости со старыми приложениями.

Однако системы на основе Linux до сих пор не преодолели даже двухпроцентный барьер установок на настольных компьютерах, хотя доминируют на рынке мобильных устройств и серверов.

Из всего сказанного приходится сделать вывод: Linux уже давно дружелюбен к пользователю, но до сих пор не добился дружелюбия к разработчику.

## Все дело в моделях финансирования?

На недавно проходившей конференции GUADEC 2012 в числе прочих обсуждаемых тем разработчики из компании Yorba Адам Дингл [Adam Dingle] и Джим Нельсон [Jim Nelson] представили новые способы коллективного финансирования разработки приложений.

Одна из инноваций в области коллективного финансирования – подход «плати сколько хочешь или не плати совсем». На нем основан *Humble Bundle*, эксперимент по продажам сборников игр. Другой подход – финансирование сообществом [crowd funding]. Основной игрок здесь *Kickstarter*, хотя существуют и другие. Проект на *Kickstarter* может выложить любой, но проекты одобряются выборочно. После апробации каждая кампания определяет временные рамки и базовую цену, а деньги присуждаются по принципу «все или ничего» по ее окончании.

По словам Дингла, настало время создания нового сайта для коллективного финансирования программного обеспечения. Этот сайт надо строить с учетом как положительного опыта *Kickstarter*, так и причин неудач других проектов. В компании Yorba эта идея обсуждается, и в идеале такой сайт должен иметь простой дизайн и быть внешне привлекательным, как *Kickstarter*, но специализироваться только на ПО (и, возможно, только на свободном ПО).

При создании сайта потребуются изменить подход к некоторым принципам, чтобы четко соответствовать потребностям проектов, создающих свободное ПО.

Открытым остается вопрос, будут ли спонсоры поддерживать разработку какой-то функциональности или же разработчики будут определять «целевую» функциональность на разных уровнях.

# softline®

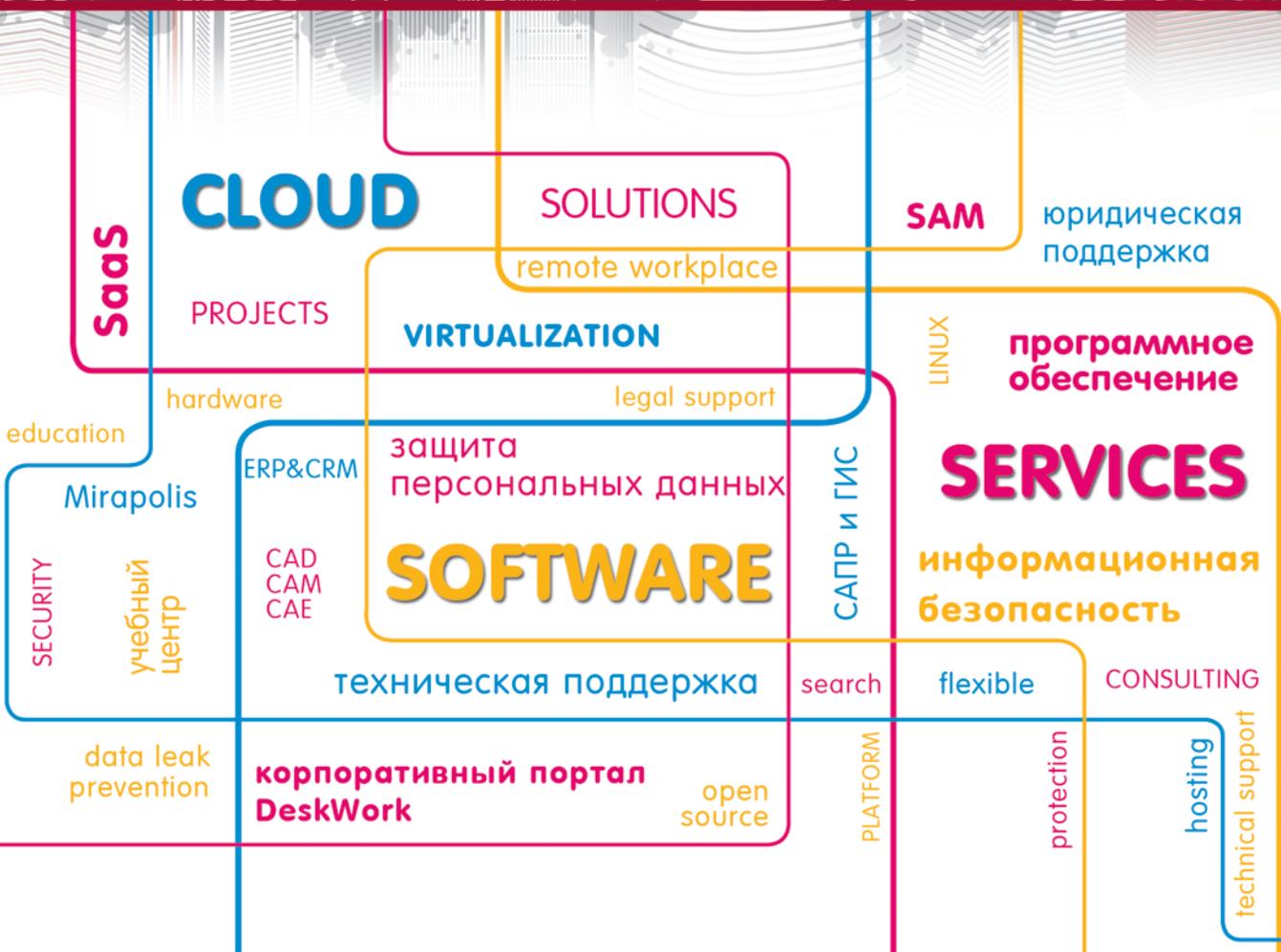


Services

Software

Cloud

## ИТ-архитектура вашего бизнеса



СПО — ДЛЯ ДОМА, ДЛЯ СЕМЬИ

# Жилье с интеллектом

openHAB 1.0 — платформа для обеспечения работы систем домашней автоматизации.

После двух с половиной лет разработки состоялся выпуск проекта openHAB (open Home Automation Bus), создавшего открытую платформу для организации работы «умного дома», способную, при установке надлежущего оборудования, выполнять такие функции, как включение/выключение освещения, подача напряжения на розетки и открытие/закрытие жалюзи на окнах. Код проекта написан на языке Java и оформлен в виде модульной системы OSGi (Open Services Gateway Initiative). OSGi — это спецификация динамической модульной шины для создания Java-приложений, разрабатываемая консорциумом OSGi Alliance. Круг ее применений довольно широк. Изначально она разрабатывалась для встроенных систем (в частности, для автомобилей BMW; активное участие принимала также Siemens), но сейчас на базе OSGi строят многофункциональные приложения (например, Eclipse SDK) и Enterprise-системы. Распространяется под лицензией GPLv3. Для организации работы web-интерфейса задействован Jetty — свободный контейнер сервлетов, целиком написанный на Java.

OpenHAB предоставляет средства для организации шины, обеспечивающей согласованную работу различных систем, оборудования и интерфейсов домашней автоматизации, позволяя передавать через данную шину команды и получать информацию о состоянии. При этом openHAB не зависит от протоколов и оборудования, предоставляя отдельный уровень абстракции, позволяющий взаимодействовать с различными типами устройств и ПО. Для определения управляющей логики предлагается использовать скрипты, написанные на специальном предметно-ориентированном языке программирования, разработанном при помощи языка программирования Eclipse Xtent.

Поддержка различных устройств и протоколов реализуется через специальные подключаемые классы — например, подобные классы созданы для поддержки протоколов управления домашней автоматизацией. Сейчас в openHAB доступна поддержка таких стандартов, протоколов, технологий и продуктов, как KNX (сетевой протокол для автоматизации зданий), HTTP, SNMP, NTP, WoL (Wake-on-LAN), Bluetooth, маршрутизаторы AVM Fritz!Box, VoIP-система Asterisk, музыкальный плей-



ер MPD, приложение для записи видео VDR (Video Disk Recorder).

Управление openHAB происходит через web-интерфейс, в котором допускается подключение дополнительных виджетов (например, виджет для включения/выключения света в каждой комнате или виджет для просмотра видео с камер слежения). Для интеграции со сторонними приложениями предоставляется управляющий REST-API. Дополнительно подготовлено несколько альтернативных систем управления, таких как клиентские приложения для платформ Android и iOS, позволяющие управлять системой с мобильного телефона или планшета, а также оперативно получать уведомления о событиях. Интересно, что для взаимодействия с сервером можно использовать протокол XMPP, что в конечном итоге позволяет пользователю присоединиться к openHAB при помощи Jabber-клиента или GTalk, и передать необходимые команды в форме «чата» со своим домом.

Среди прочих полезных функций отмечается поддержка интеграции с календарем-планировщиком Google Calendar, через который пользователь может задавать события, намеченные для исполнения системой домашней автоматизации в указанное время. Например, можно добавить в планировщик событие, по которому каждый вечер будут закрываться жалюзи или временами включаться свет для имитации присутствия людей в доме. Также поддерживается хранение конфигурации, управляющих команд и логов в облачном хранилище Dropbox, что позволяет изменять настройки и просматривать логи из любой точки сети. На основе создаваемых логов система поддерживает создание наглядных диаграмм и графиков, по которым, например, можно судить об изменении энергопотребления в указанный период времени.

Исходные коды openHAB доступны под свободной лицензией GNU GPLv3. Подробности проекта можно узнать на сайте <http://code.google.com/p/openhab/>. Там же можно найти инструкции по разработке для openHAB с помощью Eclipse IDE. **Л.ХФ**

## Новости короткой строкой

» Компания Twitter вступила в некоммерческую организацию Linux Foundation.

Источник: [www.arstechnica.com](http://www.arstechnica.com)

» В новых смартфонах компании Jolla, основанной бывшими сотрудниками Nokia с целью разработки новых устройств на основе Linux-платформы MeeGo, будет возможность устанавливать, запускать и выполнять приложения для Android.

Источник: [www.gigamon.com](http://www.gigamon.com)

» Компания «Доктор Веб» сообщила, что для Linux и Mac OS X появился принципиально новый бэкдор под названием BackDoor.Wirenet.1 с функцией кейлогера, которая позволяет отслеживать нажатия клавиш в ряде популярных приложений, таких как Opera, Firefox, Chrome, Chromium, Thunderbird, SeaMonkey и Pidgin.

Источник: <http://news.drweb.com>

» В Рунете заработало открытое сообщество [wasp-guard.ru](http://wasp-guard.ru), призванное создать масштабируемую распределенную систему обнаружения вторжений, которой сможет пользоваться любой системный администратор в любой части света. В проекте используются только открытые и (или) свободные инструменты мониторинга, контроля, обнаружения и устранения вторжений.

Источник: [www.wasp-guard.ru](http://www.wasp-guard.ru)

» 14–16 сентября 2012 г. в Одессе в Центре суперкомпьютерных вычислений и свободного программного обеспечения ОНУ им. И. И. Мечникова при активном участии компании RootUA прошла V Международная конференция для профессионалов FOSS Sea 2012.

Источник: [www.foos-sea.org.ua](http://www.foos-sea.org.ua)

» SUSE Linux Enterprise Server 11 SP1 получил сертификат ФСТЭК по 5-му классу защищенности и 4-му классу НДВ.

Источник: [www.fstec.ru](http://www.fstec.ru)

» Петербургская компания CSBI Group на конференции CNews представила свободную автоматизированную банковскую систему (АБС). Ее код будет поставляться заказчикам бесплатно.

Источник: [pda.cnews.ru/news](http://pda.cnews.ru/news)

» Лидер отечественного рынка устройств для электронного чтения, компания Покетбук, представила две новые бюджетные модели ридеров-читалок.

Источник: [www.pocketbook-int.com/ru](http://www.pocketbook-int.com/ru)

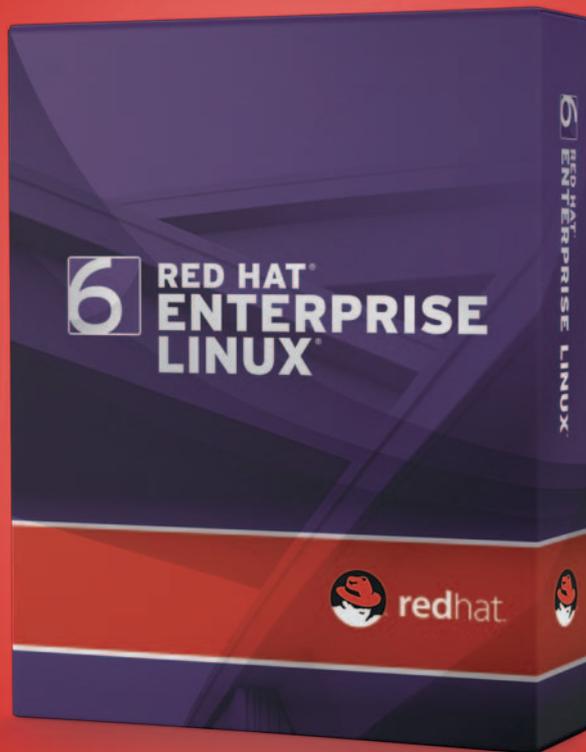
# Red Hat Enterprise Linux

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

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

## Red Hat — ведущий серверный дистрибутив Linux

- Более 15 лет промышленного использования
- Свыше 80% рынка корпоративного Linux по данным CIO Insight
- 5 лет среди лучших вендоров
- Выгодная совокупная стоимость владения (TCO)
- Поддержка в течение 10 лет

### Специальное предложение!

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



Москва  
+7 (499)

271-49-54

Санкт-Петербург  
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. [www.linuxcenter.ru](http://www.linuxcenter.ru)

Linux  center

# День свободы ПО 2012

Москвичи встретили этот день митингом в центре города, рассказывают Дмитрий Шурупов и Юрий Татарский.

**15** сентября наши коллеги и энтузиасты со всего мира отмечают День свободы программного обеспечения – SFD (<http://softwarefreedomday.org/>). Традиция этого праздника зародилась 8 лет назад: с тех пор сторонники свободного ПО ежегодно собираются и проводят совместные мероприятия для продвижения своих идей, обсуждения насущных вопросов и просто приятного общения. В рамках события организуются конференции, семинары или InstallFest'ы – в общем, выбираются привычные для сообщества форматы. Однако в этом году мы решили поступить иначе...

Загадочное местоимение «мы» скрывает три компании – партнерские агенты Canonical в России: «Флант», Progressive Operating Systems и POWERCOMP. Весной они проводили московский Ubuntu InstallFest 12.04, по итогам которого решили продолжить продвижение свободного ПО в массах, и, обсуждая SFD 2012 с Московской группой пользователей Linux (MLUG), пришли к выводу, что хорошо бы сделать событие, способное привлечь не только приглашенных людей из сообщества, но и стороннюю публику.

Но как это сделать? Нам помогла встреча с Павлом Рассудовым – председателем «Пиратской партии России». Несмотря на зловещее название, партия выступает за свободные лицензии и свободное ПО (<http://pirate-party.ru/node/5>). А по роду своей деятельности они имеют опыт проведения уличных мероприятий. И Павел сразу же предложил организовать митинг в центре города. Сначала мы ступешались: какой еще «митинг», если речь идет об ИТ-праздновании? Но, обсудив возможные варианты организации, поняли, что связь слова «митинг» с политикой – просто сложившийся в обществе стереотип. А по сути, это встреча («meeting») единомышленников.



► Единомышленников можно разыскивать и таким способом...

## Празднование

Первым делом мы продумали раздаточный материал. Во-первых, это, конечно же, Linux-дистрибутивы: CD с Ubuntu мы подготовили сами, плюс удалось привлечь сообщество Russian Fedora, а также разработчиков отечественной ROSA Marathon, потомка Mandriva. Поддержали нас и коллеги из ГНУ/Линуксцентра, специально для мероприятия выделившие журналы «Linux Format» и диски к журналу. Кроме того, мы напечатали два типа буклетов:

» с общими сведениями о свободном программном обеспечении и Дне свободы ПО;

» с информацией про Ubuntu и ссылками на популярные в России web-ресурсы.

В Новопушкинском сквере мы установили звуковое оборудование и сцену, украшенную фирменной атрибутикой от организаторов SFD (баннер и воздушные шары заказывали одновременно у американских коллег). Само мероприятие состояло из выступлений членов русскоязычного СПО-сообщества и развлекательных сцен в исполнении актеров.

Выступлений планировалось десять: Юрий Татарский (Progressive Operating Systems, [yury.tatarsky@p-os.ru](mailto:yury.tatarsky@p-os.ru)) рассказал о празднике, организаторах, выступающих, а Павел Рассудов – об отношении Пиратов к СПО. EQuivalent (MLUG) задал полемический вопрос «Принадлежат ли ваши программы?». XMs (MLUG) описал лицензии на софт, CYB3R (MLUG) – тивоизацию, DRM и «vendor lock-in». Кирилл Лебедев (POWERCOMP) озвучил проблемы противостояния свободного и проприетарного ПО на рынке. Об особенностях сообщества говорили Петр Леманков (Russian Fedora) и Станислав Погоржельский (Russian Ubuntu LoCo Team). Valber (MLUG) напомнил об успехах СПО в науке, а Дмитрий Шурупов (Флант, [dmitry.shurupov@flant.ru](mailto:dmitry.shurupov@flant.ru)) охватил диапазон от философии до бизнеса. Речь умышленно делалась пятиминутными, чтобы зрители не заскучили. Во время празднования к нам подошел украинский коллега Михаил Шигорин из ALT Linux Team и тоже попросил слова. Немного времени у нас в запасе было, и мы с удовольствием согласились.

Выступления «разбавлялись» сценками из мини-сериала про компьютерный сервис. По сюжетам, придуманным нами и членами MLUG, режиссер Алексей Себреников написал небольшой сцена-



► Антилопа-гну в очередной раз выступает героем лозунга на службе прогресса.

рий, и его воплотили актеры Николай Кобров и Юлия Галина. Зарисовки отражали проблемы среднестатистического пользователей проприетарного ПО, и их решение с помощью свободного ПО, и были сделаны с юмором, так что публика встретила их здоровым смехом.

Кстати о публике: из-за намечавшегося по соседству политического митинга («Марш миллионов») полиция приложила максимум усилий, чтобы оградить наше мероприятие от провокаторов. Многие прохожие опасались к нам проходить; но в итоге митинг посетили около 200 человек (что мы оценили по числу розданных дисков с Linux-дистрибутивами). После мероприятия энтузиасты из MLUG взяли часть оставшихся материалов и раздали их прохожим на улицах Москвы.

## Не Москвой единой

Software Freedom Day 2012 отмечался и в других городах России. В особенности отличились Саратов (<http://habr.ru/post/151539/>), Ульяновск (<http://ul.sfd2012.ru/>), Омск (<http://bit.ly/sfd2012omskannounce>) и Новосибирск (<http://lugnsk.org/sfd2012>). Полный список городов этим не ограничивается – см. <http://wiki.softwarefreedomday.org/2012/Russia>. **EXF**

## Спешите видеть!

Все фото с Праздника доступны на <https://plus.google.com/photos/102135447748820605505/albums/5788520453323030017>. Видеозапись с отчетом о мероприятии и интервью посетителям можно посмотреть на <http://youtu.be/XBBHaEpUebE>. А все праздничные видео размещены на канале <http://ubuntu.ru>.



**АЛЕКСЕЙ ФЕДОРЧУК**  
Тэг «сарказм»  
по умолчанию,  
смайлики по вкусу.

## openSUSE 12.2: детектив вокруг релиза

**К**огда вы будете читать этот номер, выход указанного релиза состоится как факт. Но ему предшествовала почти детективная история. 31 августа «по России слух прошел», что образы долгожданного релиза – на официальных зеркалах проекта. Правда, они почти мгновенно исчезли, но остались доступными несколько уже неофициальных зеркал, организованных предприимчивыми студентами. Но через день и они таинственным образом исчезли. Возникает вопрос: не была ли это сознательная утечка информации со стороны разработчиков? Цель которой – развлечь народ, пока они отдыхают от трудов праведных и сочиняют текст пресс-релиза. Что делает честь их чувству юмора.

Однако заинтересованные лица (включая автора этих строк) успели вожаделенные образы скачать и даже установить с них систему, получив шанс ознакомиться с еще не объявленным релизом досрочно. Система, с них установленная, оказалась более чем работоспособной. А вслед появились уже и официальные репозитории, так что не было проблем, скажем, с сетевой установкой или обновлением кандидата в релизы.

Что же до собственно релиза – он вполне оправдал возлагавшиеся на него надежды. А именно: в нем нет ничего ультра-революционного. И по-прежнему существует выбор между стилем «гипермодерн», причем в очень смягченном варианте, и традиционным стилем, вот уже более двух десятилетий обеспечивающим openSUSE привязанность пользователей.

alv@posix.ru

## Сегодня мы рассматриваем:

- Opera против Chrome...** ..... 14  
Этот браузер первым дал нам номера быстрого набора и вкладки, и он по-прежнему быстрее всех, но жестоко недооценивается пользователями Linux. Стоит ли терпеть проприетарную лицензию ради его замечательных функций?
- Alien Arena: Перезагрузка 7.60** ..... 18  
Возьмите пришельцев из недооцененных «Атак Марса» Тима Бертона, добавьте открытый игровой движок, веселый Quake-подобный захват флага и смертельную схватку, и вы получите эту жемчужинку среди игр. Причем с онлайн-режимом!
- ...и Chrome против Opera** ..... 14  
Google ободрал свой браузер *Chrome*, чтобы дать нам нечто подобное оригиналу – *Firefox*. *Chrome* быстр, прост и изобилует дополнениями, если они кому нужны. И как же он смотрится на фоне *Opera*?
- Skype** ..... 19  
*Skype* позволяет нам говорить с далекими близкими бесплатно, где бы они ни находились (если у них приличная полоса пропускания). Но не позволяет изучить свой код, а это не есть хорошо.
- Raspbian** ..... 17  
Это не дистрибутив, разработанный хакерами с греческого острова Респбос, а версия Debian, специально предназначенная для максимального использования красот Raspberry Pi. Мы его уже полюбили.

### Raspbian c. 17



➤ Неказистый, но полный хороших намерений вариант Debian с LXDE заточен под Pi.

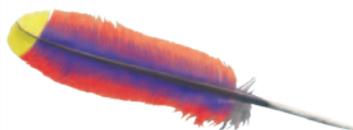
### Alien Arena c. 18



➤ После распада СССР слово «пришельцы» перестали быть метафорой. Они просто пришельцы.

## Сравнение: Лицензии на свободное ПО с. 28

Apache



BSD



GPLv2



GPLv3



LGPL



Лицензия, под которой вы выпустите свой код, капитально повлияет на успех проекта – LibreOffice это доказал. Так что прочтите все и выбирайте тщательно!

# Chrome против Opera

Ваш браузер – уже не просто браузер: это почтовый клиент, файловый сервер и много чего еще. **Адам Оксфорд** следит за битвой двух тяжеловесов.

## Вкратце

» Два полнофункциональных браузера с галерей расширений. См. также: *Firefox*, *Epirhany* (он же – *Web*) и *reKonq*.

**С**огласитесь: когда речь заходит о web-браузерах, мы так непостоянны. Хотя мы все согласны насчет убожества сами знаете чего, дай нам только выбор между *Firefox* и *Chrome* от Google – не говоря уж о своре их родителей – и прощай, верность.

А как иначе объяснишь массовую перебежку поклонников старой доброй открытой Mozilla на сторону юного *Chrome/Chromium* несколько лет назад? Ключев на скорость и минималистский шарм ставленника Google, они без сожалений отбросили наработанные схемы расширяемого *Firefox*.

И *Chrome/Chromium* отнюдь не намерен захапнуть. По данным [w3schools.com](http://w3schools.com), он уже признан самым популярным в мире браузером, и последние 3 года его аудитория стабильно увеличивается где-то на 1% в месяц. Это значит, что *Chrome* сейчас стоит практически на каждом вто-

ром ПК, и по мере роста доходов Google у всех остальных они падают.

Точнее, у всех, кроме *Opera* – сумевшего набрать небольшую, но стойкую армию пользователей. Итак, резвый юнец против дотошного старца; чью сторону примете вы? Выясним это, сравнив *Chrome 20* и *Opera 12*.

## Браузер от Google

Для проведения этого теста мы выбрали *Chrome 20*: именно он – официальный продукт Google, а не практически идентичный ему *Chromium* (собственно говоря, родитель вышеупомянутого). Напомним, что *Chromium* – открытый браузер на базе *webkit*, созданный при поддержке Google и послуживший основой для *Chrome*. *Chrome* дополнительно содержит встроенный инструмент для чтения PDF и Flash-проигрыватель, а также утилиту отслеживания активности использования

программы (*RLZ*); других отличий не имеется. *Chromium* чаще встречается в репозиториях дистрибутивов, хотя, как правило, не в последней версии. Например, в Ubuntu 12.04 на нашей тестовой машине по умолчанию оказалась доступна только версия 18.

*Chrome 20.0.1132.57* мы установили из DEB-файла, который скачали с домашней страницы проекта. И первое, что бросилось в глаза – насколько *Chrome 20* неотличим от первой версии *Chrome*. Старое правило «Не сломалось – не чини» бережно соблюдается. Ни намек на изменение привычной функциональности. При первом прогоне, как было и раньше, не появляется почти ничего: только единое поле URL/поиска (оно же – мультибар) и иконка управления, вызывающая аналог меню Файл.

По умолчанию, *Chrome* избегает украшательства, оставив только узкую пря-

## Как отличить Opera (слева) от Chrome (справа)

### Кнопка меню

Новый стиль *Opera 12* напоминает обновленный *Firefox* и... *Chrome*! Разница только в сглаженных прямоугольных формах и фирменной кнопке меню.

### Мультибар

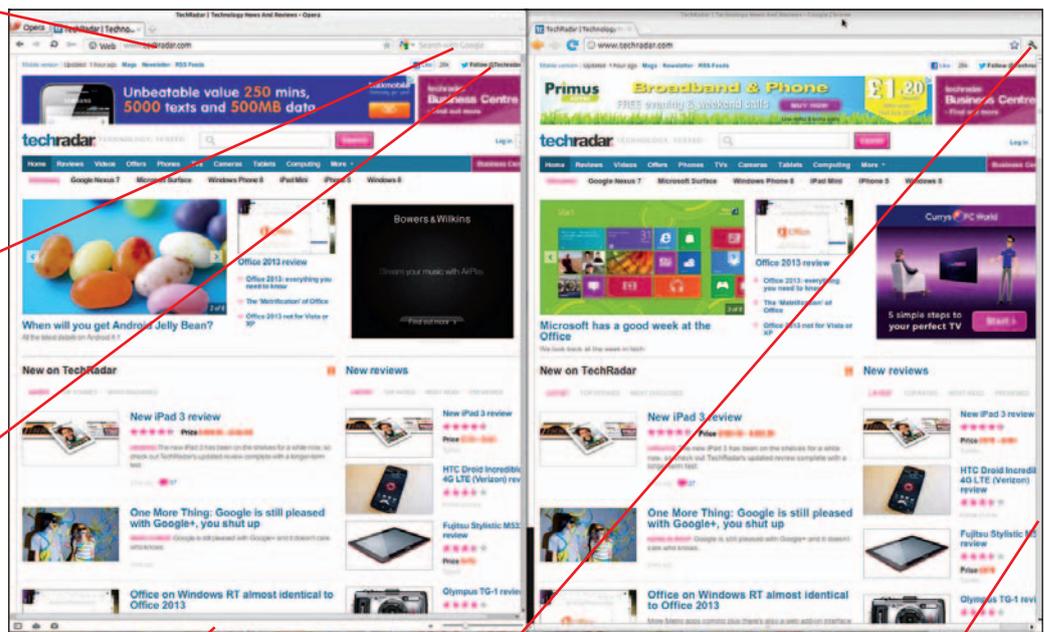
В *Opera*, помимо него, есть и отдельная панель поиска. Можно вводить и в нее, и в основную.

### Текст

Приглядитесь повнимательнее, и вы заметите разницу в текстовых подсказках. В *Chrome* шрифт более красивого начертания.

### Отображение Java

Скорость в данном тесте – по-прежнему козырь *Chrome*. А вот новый HTML5 – уже другая история.



### Нижняя панель

В *Opera* видима по умолчанию, содержит некоторые команды быстрого доступа и линейку масштаба.

### Настройки

В *Chrome* они спрятаны под иконкой настройки (гаечный ключ). Настройка, впрочем, далеко не заходит.

### Полосы прокрутки

Присутствуют в обоих браузерах. Но не такие элегантные, как исчезающая в Ubuntu.

моугольную голубую строку заголовка, которая сменится на меню Параметры системы, если правой кнопкой щелкнуть в верхней части экрана. Также по желанию можно отобразить под мультибаром панель Закладок.

Кое-какие перемены внешности все же есть: шрифт в Настройках стал поуже, да и в целом макет страницы стал еще более радикально прост. И, к сожалению, лишен юмора: вместо меню верхнего уровня, где обычно прячут системные и пользовательские параметры – прозаичные и ненамного более удобные Настройки, из которых можно попасть в Дополнительные. Хотя, конечно, все четко и понятно. Любая из немногочисленных настроек достижима за пару самоочевидных щелчков; только опция Не отслеживать печалит своим отсутствием.

## Работа над ошибками

В 20-й версии *Chrome/Chromium* исправлено много ошибок, включая наиболее распространенные помехи с Google Cloud Print – сервисом, позволяющим виртуально получать доступ к любым принтерам, используя *Chrome*.

Можно, конечно, прекрасно обойтись и без *Chrome* – многие, по опыту своего общения с Google, так и сделают. Если вы не из их числа, то вам будет достаточно знать пароль, найти нужную вкладку и иметь встроенную функцию синхронизации, которая теоретически позволяет объединять Google'овский *Android Browser* и *Chrome*. На практике же он ведет себя довольно капризно, особенно если у вас много открытых вкладок и устройств на Android.

С другой стороны, это не проблема, ведь феерическому взлету *Chrome* сопутствовал и активный рост онлайн-магазина всевозможных расширений для этого браузера. Их качеству и количеству любой другой браузер может только позавидовать. Соз-



► В фокусе *Opera*: некое солидное, всемирно известное учреждение. Ну и сайт Би-би-си заодно.

данный по модели Apple App Store или GooglePlay, он включает и лучшие инди-игры, такие как *Bastion*, *DragonAgeLegends* и *AngryBirds*, и читалки лент новостей, и органайзеры, и лучшую реализацию *Evernote* на сетевой платформе. *LastPass* и *Xmarks* куда лучше встроенных паролей и синхронизированных закладок. А с помощью *Chrome* (но не *Chromium*) можно даже запустить Google Docs в офлайн-режиме. Все это работает с ускорением, поддерживаемым вашей системой, и использует мощь вашей видеокарты.

Касаемо расширений, у *Chrome* всего один недостаток: то, что они разделяются на два типа. Те, что связаны с работой самого обозревателя, всегда доступны и отображаются на панели инструментов с мультибаром, другие – например, игры – запускаются со стартовой страницы браузера.

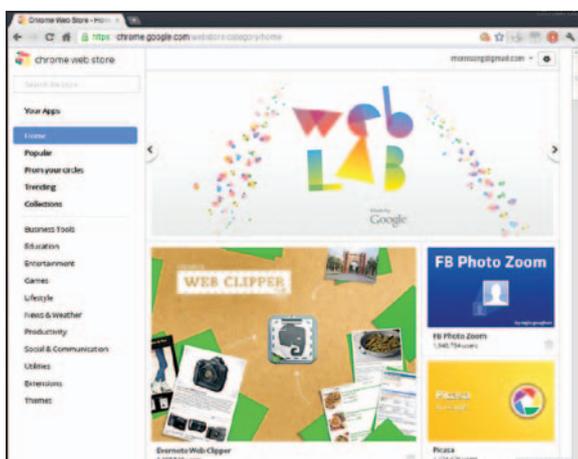
## Новое лицо соперника

В то время как *Chrome* усиленно сопротивляется переменам, *Opera 12* за последние

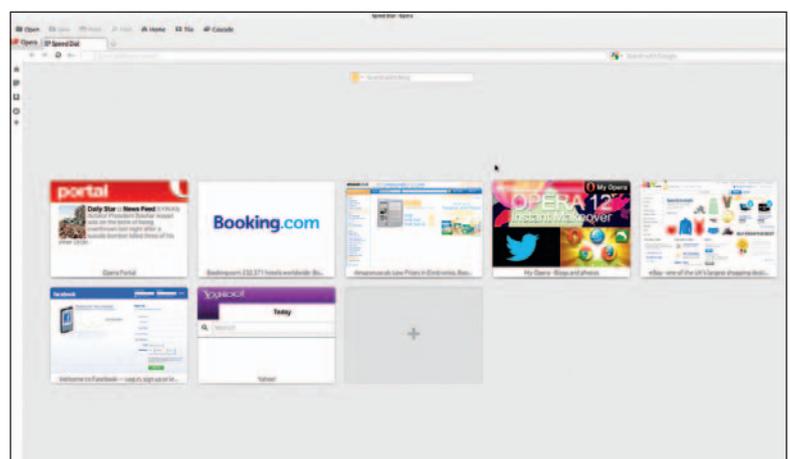
пара лет изменилась до неузнаваемости – столь радикально, что исходный интерфейс ее теперь похож, ха-ха, на *Chrome*. По умолчанию все лишнее отключено, максимум места представлен открытой web-страницей. Верхняя часть с закладками поражает сходством; разве что здесь есть еще пара кнопок, например, для изменения фонового рисунка, а также внизу присутствует нижняя панель.

В отличие от *Chrome*, в *Opera* масса меню и панелей навигации, включаемых по вашему усмотрению. Можно добавить боковую панель для быстрого доступа, истории и т.д., а сверху – панель главного меню, при помощи которой вкладки можно располагать каскадом подобно тому, как это делается с окнами.

Как и в *Chrome*, в *Opera* есть специальная вкладка, пароль и функция синхронизации, называемая здесь *Opera Link*, и у нее те же недостатки, что и у аналогичного сервиса Google: если у вас не стоит *Opera* на всех рабочих местах и мобильных устройствах, толку от нее будет не боль-



► Web Store от Google не устает пополняться приложениями, затмевая все прочие браузеры.



► Только попробуйте копнуть глубже – и вы убедитесь, что в *Opera* настроить можно абсолютно все.



## Шаг за шагом: Включаем скрытые настройки



### 1 Добираемся до настроек

В обоих браузерах есть дополнительные настройки. В Орега добраться до них можно, набрав `opera:config` или нечто подобное в окне URL. Так, чтобы включить аппаратное ускорение, набираем `opera:config#UserPrefs!EnableHardwareAcceleration` и меняем флажок на 1.

### 2 Включаем WebGL

По умолчанию, разработчики Орега отключили WebGL. Его используют немногие приложения, но с приходом в браузеры игрового движка Unity полезно хотя бы знать, где находится эта настройка: `opera:config#UserPrefs!EnableWebGL` и переключаем в 1.

### 3 Активируем функции

В *Chrome* тоже можно включить дополнительные функции или отключить аппаратное ускорение, зайдя на `chrome://flags`. Правда некоторые настройки довольно нелогичны: например, флажок *WebGL* почему-то отключает функцию, а не включает ее.

ше, чем от стороннего приложения. К сожалению для *Opera*, число расширений у этого браузера намного скромнее, и если нехватка игр – не такая уж проблема, то отсутствие популярных приложений типа *Evernote* или *Xmarks*, вероятно, вовсе работающих на вашем мобильнике или планшете, уже серьезнее.

## «В Орега нет аппаратного ускорения по умолчанию.»

В удобстве использования браузеры практически одинаковы: Орега выигрывает благодаря встроенным функциям для индивидуальной настройки, но проигрывает по части доступных приложений.

Ну, а как же оба ведут себя в деле? В Орега нет аппаратного ускорения по умолчанию, и эта функция далеко упрятана от новичков (см. Шаг за шагом). А жаль, потому что она резко улучшает

отображение HTML5. При базовом составе, *Chrome* значительно превосходит *Opera* в работе Java и выполнении операций с CSS, хотя в целом отдать преимущество какому-то одному из браузеров сложно. А вот в интерактивных тестах HTML5 отличия разительны: *Chrome* задействует возможности вашей видеокарты, проходя задания быстро, сглаживая частоту видеокадров, так что HTML5 ощущается как родной, хотя есть проблема с проприетарным драйвером NVIDIA, отключающим ускорение при обработке 2D Canvas.

### Ускоряемся

*Opera* полагается только на процессор и быстро выдыхается при обработке страниц со сложной структурой. По показателям производительности ему не догнать *Chrome*, даже если у того нет доступа к видеокарте. Но стоит добраться до потайных настроек ускорения, как 3D-эффекты и WebGL в *Opera* буквально перерождаются, моментально догоняя и даже обгоняя *Chrome* (хотя и не во всем).

*Opera* потребляет меньше памяти, чем *Chrome*, который держится за свою уникальную функцию запуска каждой вкладки и web-приложения отдельным процессом. Это, конечно, захламляет системный монитор, но стабильность гарантируется при любом количестве открытых вкладок.

*Chrome* также прошел большее число тестов безопасности *Browserscope*, нежели *Opera*. И, по нашему мнению, текстовые подсказки на движке *Webkit*, реализованные в *Chrome*, смотрятся гораздо лучше, чем в *Opera*, где в основе – *Presto*. Так что же выбрать? Если вы давно не пользовались *Opera*, то попробовать стоит, особенно с аппаратным ускорением. Перегруженный интерфейс и мизерные шрифты ушли в прошлое, и браузер вы-

глядит современно, да вдобавок индивидуально настраивается.

И все-таки мы рекомендуем *Chrome 20*, а точнее – открытый *Chromium 20*. Это по-прежнему лидер и по скорости, и по безопасности. Приплюсуем сюда превосходное качество расширений, и поле останется за ним. По крайней мере, пока. **LXF**

LINUX FORMAT Вердикт

**Chrome 20**

Разработчик: Google/Chromium Project  
 Сайт: [chrome.google.com](http://chrome.google.com)  
 Цена: Бесплатно

|                    |      |
|--------------------|------|
| Функциональность   | 9/10 |
| Производительность | 9/10 |
| Документация       | 5/10 |

» Не зря *Chrome* считается номером один, и все ему подражают. Это по-прежнему самый простой и быстрый браузер из всех.

Рейтинг 8/10

LINUX FORMAT Вердикт

**Opera**

Разработчик: Opera Software  
 Сайт: [www.opera.com](http://www.opera.com)  
 Цена: Бесплатно

|                    |      |
|--------------------|------|
| Функциональность   | 8/10 |
| Производительность | 8/10 |
| Документация       | 7/10 |

» По скорости и безопасности *Opera* на втором месте почти без отрыва; подводит только закрытая лицензия и недостаток расширений.

Рейтинг 8/10

## Результаты тестов

|   | Chrome 20 | Opera 12 |
|---|-----------|----------|
| Sunspider 0.9.1, мс                     | 194,8     | 278,3    |
| Потребление памяти (40 вкладок)         | 1,312 ГБ  | 742 МБ   |
| Maze Solver, с                          | 4,3       | 11       |
| Тест HTML5, кадр/с (www.craftymind.com) | 31,25     | 52,84    |
| WebVizBench, кадр/с                     | 20,12     | 29,01    |
| Загрузка страницы, с                    | 1,723     | 1,2      |
| Аквариум WebGL (50 рыбок), кадр/с       | 58        | 50       |

# Raspbian

У нашего малыша-компьютера появился новый дистрибутив. Бен Эверард отодвигает пиццу и принимается за Pi другого рода.

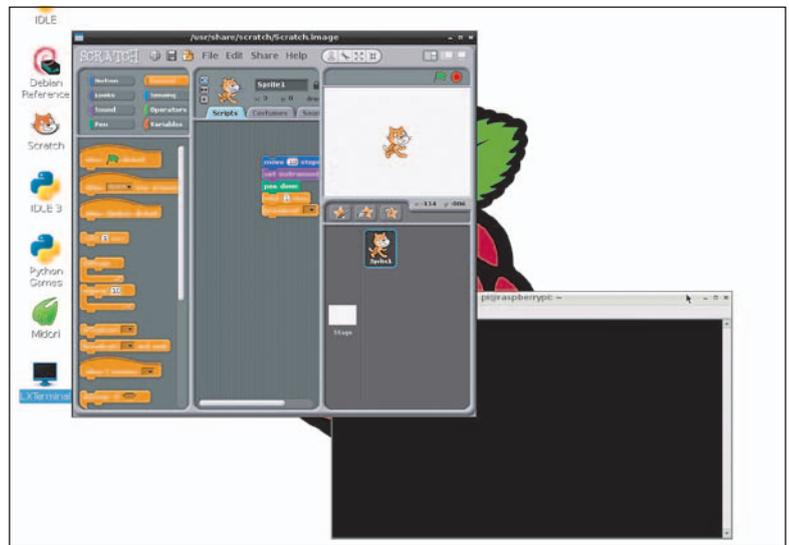
## Вкратце

» Raspbian — это версия Debian Wheezy, созданная специально для Raspberry Pi. См. также: Arch Linux Arm и QtonPi.

**Н**а случай, если вы провели последние полгода, засунув голову в ведро с песком, сообщаем, что Raspberry Pi — это недорогой компьютер на основе ARM. Из-за своего оборудования он нуждается в особых дистрибутивах. Таковых уже несколько, но Raspbian недавно получил гордое звание Рекомендованного дистрибутива от команды Raspberry Pi.

За графический интерфейс выбран *Lightweight X11 Desktop Environment*. У LXDE немало фанатов — если верить цифрам Distrowatch, среди клонов Ubuntu лидирует Lubuntu. Ваш покорный слуга к фанатам не принадлежит. Интерфейс выглядит сляпанным из несоразмерных частей — одни лаконичны, другие громоздки. Raspbian тут ни при чем, но такой уж у него выбор по умолчанию. Тяжеловесы вроде KDE и Gnome на таком оборудовании не заплещут, но вот нечто полегче, вроде Xfce, вполне сгодится. Xfce можно найти в репозиториях Raspbian и легко установить, набрав `sudo apt-get install xfce`. Правда, запустить его без экранного менеджера вы сумеете не сразу (подсказка: наберите `startxfce` в текстовом режиме).

В составе программ есть огромные прорехи — предположительно, потому, что команда пыталась дать необходимый минимум, в который уже сам пользователь добавлял бы, что ему нужно. Раз так, то отсутствие графического пакетного менеджера вдвойне удивительно. Raspberry Pi ориентирован на новичков, и можно ли рассчитывать, что они будут использовать `apt-get` в командной стро-



» Рабочий стол LXDE функционален, но буквально ошарашивает недостатком лоска.

ке, чтобы собрать себе базовый функционал? У тех, кому Raspberry Pi нужен для работы в группах, голова об этом болеть будет меньше. Поскольку копировать SD-карты — не проблема, достаточно создать для себя идеальную сборку, а потом ее растащить.

## Рост производительности

Основной целью команды Raspbian были вовсе не графические изыски. Они стремились, чтобы программы были под стать оборудованию. И это первый дистрибутив, который реально задействовал вычисления с плавающей запятой. Звучит не слишком волнующе, но в результате значительно улучшена производительность многих приложений — включая web-браузеры. При сравнительном тестировании разработчики обнаружили, что аппаратное ускорение разогнало приложения в среднем на 18 %.

Процесс установки также перенастроен, чтобы все шло более гладко с первого же запуска. Теперь легко увеличить образ, чтобы он целиком заполнил SD-карту — при помощи всего одного хитрого шага. Можно также запустить SSH-сервер, изменить границы разделов или выбрать другой часовой пояс.

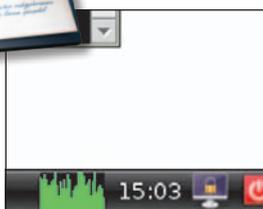
Предустановленные программы сконцентрированы на обучении программированию. Scratch предоставляет графиче-

скую среду, чтобы втянуть детишек в программирование, тогда как IDLE (IDE Python) — уже более сложный инструмент кодера.

Взвесив разные аспекты этого дистрибутива, мы ощутили раздвоение личности. Наш внутренний технарь одобряет работу, проделанную командой над Pi. А вот как пользователи мы не в восторге от выбора программ и рабочего окружения. Но поскольку это пока что бета-версия для разработчиков, побеждает все же первый. Хотя мы надеемся, что Pi все-таки станет дружелюбнее к пользователям, прежде чем выйти в народ. **LXF**

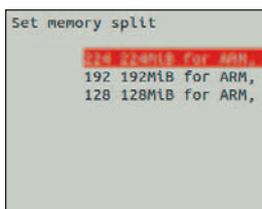


## Свойства навскидку



### Системный монитор

Этот инструмент в системном лотке следит за уровнем свободных ресурсов.



### raspi-config

Управляйте своим Pi при помощи этой удобной текстовой утилиты.

LINUX FORMAT Вердикт

**Raspbian**

Разработчик: Raspbian Project  
 Сайт: [www.raspbian.org](http://www.raspbian.org)  
 Цена: Свободно по смеси лицензий

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

» Лучший дистрибутив для Raspberry Pi на данном этапе, но надеемся на прогресс.

Рейтинг 8/10

# Alien Arena: Reloaded Edition 7.60

Большеголовые пришельцы, ветхие карты и стада коров – все это убеждает Джонатана Робертса, что *Alien Arena* – стрелялка в лучших традициях.

## Вкратце

» Стрелялка в духе *Quake* с парочкой веселых сюрпризов. См. также: *Warsow* или *Nexuiz*.

Если вам нравились *Quake III*, *Unreal Tournament* и фантастику 1950-60-х, то *Alien Arena* – для вас. Ведь в ее основе все та же формула успеха (быстрый темп, смертельные онлайн-схватки с огромными автоматами, множество карт и кровопролитие), обернутая в инопланетную кич-эстетику.

Как и *Quake II*, и многие другие свободные и открытые стрелялки от первого лица, это игра на том же старом движке *Tech 2*, выпущенная id software как свободный код. COR Entertainment за эти годы значительно улучшили этот движок, и теперь игра практически не отстает от более современных.

«Теперь игра практически не отстает от более современных.»

» На головах пришельцев держалась вся фантастика 1950-х, и мы рады снова видеть их в *Alien Arena*.

Функции движка включают попиксельное освещение, «физику тряпичной куклы» [особый тип анимации персонажа, – прим. пер.], световые ореолы, естественный свет и мягкие тени. Да, мы тоже понятия не имеем, что все это значит, но благодаря им карты, оружие и игроки чертовски хорошо выглядят.

Пожалуй, текстуры выглядят слегка уплощенными, а большинство фигур – неотесанными, но все мелькает настолько быстро, что вам некогда это замечать. Карты в основном потемневшие, ветхие и подобающе жуткие. Нормальный ход – железные катакомбы посреди зеленой радиоактивной жижи, отравляющей здоровье, и громадные конструкции, служащие прекрасным укрытием.

Важнее карт и графики, конечно, игровой процесс, и в *Alien Arena* он превосходный. Здесь есть даже одна из лучших фишек классического *Quake*: прыжки с трюками. То есть можно палить по полу и стенам, одновременно совершая гигантские прыжки. А можно и скакать, как заяц, чтобы ловчее уклониться от выстрелов противника.



» Потемневшие карты, плохое освещение и множество надежных укрытий – что еще нужно, чтобы истребить пришельцев?

Режимов игры множество – есть из чего выбрать. Естественно, доступен классический Бой насмерть [Deathmatch], но мой личный фаворит – Захват флага [Capture the Flag]. Добавлено много чего интересного: например, Массированная атака [All-out Assault] – где вы можете пилотировать все корабли, находящиеся на карте, сразу – и Электропогонялка [Cattle prod], где вы должны загнать стадо в портал в целостности и сохранности (это, пожалуй, реверанс в сторону классической фантастики про истребление скота), попутно убивая врагов. Все это разнообразие давно привешую формулу и вносит в нее прелесть новизны.

## «Уделай» друга

Как и в любую стрелялку, в *Alien Arena* нужно играть онлайн. Ведь реальные противники куда лучше ботов; а возможность играть с друзьями, заодно глумясь над ними через встроенный IRC-клиент, еще добавляет острых ощущений.

Если у вас нет друзей, с кем можно было бы поиграть, возможно, вам стоит присоединиться к чьей-то игре. Но на большинстве серверов все же в основном обитают боты, и только на некоторых ока-

залось приличное число живых игроков. Так что лучше проверить, а потом уж играть. Игра в общем забавная и стоит внимания. Мы взяли для обзора свежую версию – 7.60, и были ужасно разочарованы, не найдя ее в репозиториях дистрибутивов. Однако скомпилировать игру из исходника для Ubuntu довольно просто; в файле INSTALL указаны все необходимые зависимости, так что пусть вас это не отпугнет. LXF

## LINUX FORMAT Вердикт

### Alien Arena: Reloaded

Разработчик: COR Entertainment

Сайт: [www.red.planetarena.org](http://www.red.planetarena.org)

Цена: Стандартная версия – бесплатно, Премиум-версия – €10 за скачивание, €15 за диск

Игровой процесс 8/10

Графика 8/10

Продолжительность 8/10

Оправданность цены 10/10

» Это здорово! Соберите друзей, скотлите LAN и проведите пятничный вечер убийно.

Рейтинг 8/10

# Skype 4

Берясь за новый релиз от Microsoft, **Грэм Моррисон** вдруг невольно вспомнил о парнокопытных, для которых полеты обычно нехарактерны.

## Вкратце

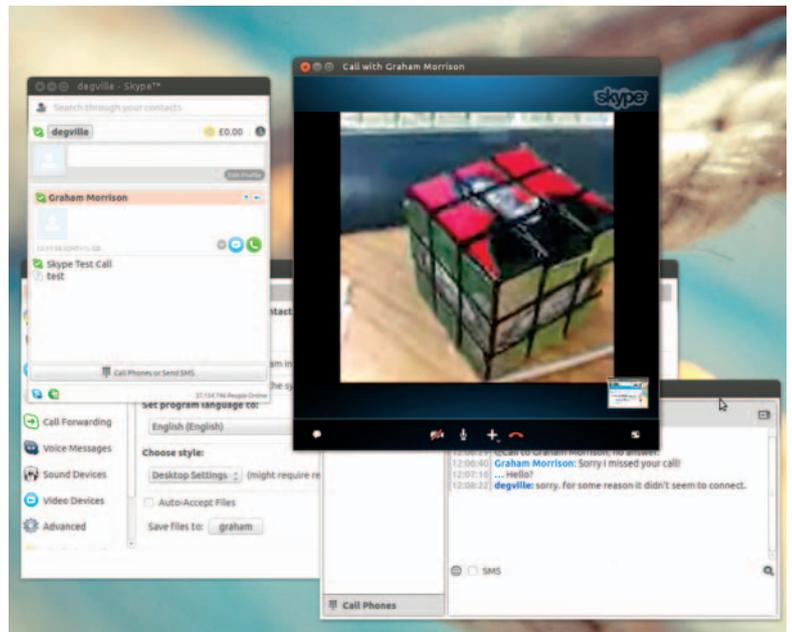
» Принимайте видеозвонки и звоните сами, в *Skype* или на обычный телефон по всему миру. См. также: бесплатный SIP-сервис.

**К**огда в 2011 мы узнали о том, что Microsoft собирается купить видеокоммуникационную компанию *Skype*, мы подумали, что клиенту Linux хана. И сперва к тому и шло, поскольку *Skype* для Linux и его релизы становились все нестабильнее. Многие тогда сочли, что оно и к лучшему: проприетарные кодеки и закрытая разработка мало созвучны с философией свободного кода, да и истории о том, что *Skype* разрешил правительственный надзор, тоже не притягивали пользователей; тем более, что есть открытые альтернативы. Но для других *Skype* был по-прежнему лучшим решением, просто потому, что он работает; и есть гордые заявления, что через него идет около 20 % всех международных звонков. В свете чего сообщение о появлении *Skype 4* для Linux нас чрезвычайно обрадовало.

4-я версия появилась спустя годы после последнего стабильного релиза и через год после дико нестабильной beta 2.2. И хотя она все равно на шаг позади OS X и Windows, но по крайней мере на уровне клиентов для Android и iOS. На самом деле, большинство наших тестов как раз и сравнивали ее с клиентом для Android, и прекрасно работали оба. Мы установили *Skype* в Ubuntu 12.04, используя пакеты для 10.04, предоставляемые *Skype*. Такое

**«Главное, не случилось ни одного падения или обрыва вызова.»**

несоответствие номеров версий обусловлено скорее скромными требованиями *Skype*, нежели политикой обновлений, поскольку вы также можете получить 64-битную версию для Ubuntu и Debian, 32-битную для Fedora 16 и OpenSUSE 12.1, плюс динамически и статически скомпонованные версии. А следовательно, команда *Skype* Linux понимает, что людям требуется. Можем также доложить, что *Skype* сразу, без всякой добавочной настройки увидел нашу USB-гарнитуру, приняв конфигурацию рабочего стола *PulseAudio*. Введя имя и подключившись, мы уже через пару минут свободно звонили и принимали звонки.



» Удивите своих друзей в другом полушарии функцией видео от *Skype*.

## Новый звук

Качество звука улучшилось, и хотя мы это отметили на фоне качества сети, переход с высокого на низкий битрейт был достаточно гладким и не мешал разговору. При нормальной скорости подключения, в *Skype* слышно лучше, чем при обычном телефонном разговоре внутри Великобритании.

Видеозвонок с телефонов Android тоже порадовал, а новое окно Видео смотрится очень приятно: собственное изображение помещено в маленькой рамке, а собеседник красуется в большой.

Наша камера Logitech работала отлично, и данный релиз обещает расширить список совместимых устройств. Также нам удалось переслать вид рабочего стола, что бывает полезно, когда кто-нибудь из двоюродных тетушек в очередной раз забыл пароль. Окно сообщений также преобразилось и выглядит лучше, а время отклика при отправке сообщений другим контактам *Skype* весьма впечатляет, особенно если это мобильная связь выше 3G. А главное, не случилось ни одного падения или обрыва вызова.

Итак, наш обзор разве что может констатировать: *Skype* просто работает. А что еще человеку нужно от средства теле-

фонии? Сеть *Skype* достаточно обширна, чтобы избрать его основной формой общения, и очень здорово, что они все-таки решили сохранить клиент Linux. Однако этот клиент защищен от декомпиляции и шлет зашифрованный трафик – причем не только во время разговора.

И это, и наличие заявлений о прослушивании говорит о том, что выбор или отказ от *Skype* сводится к спору свободы и удобства. Но по крайней мере выбор есть. Что и ценно. **LXF**

**LINUX** **Вердикт**  
FORMAT

**Skype 4**

Разработчик: Microsoft Skype  
Сайт: [www.skype.com](http://www.skype.com)  
Цена: Бесплатный Skype

|                        |      |
|------------------------|------|
| Функциональность       | 7/10 |
| Производительность     | 8/10 |
| Удобство использования | 7/10 |
| Ценность               | 7/10 |

» Прекрасно, чтобы быть на связи, но не самое лучшее по части свободы.

**Рейтинг 7/10**



## ПОРАЖЕНИЕ ГОДА

# Самый крупный в истории штраф

Компания Apple одержала победу в судебном разбирательстве с Samsung.

**В**ынесен окончательный вердикт в судебном разбирательстве Apple против Samsung. В большинстве пунктов обвинения присяжные поддержали компанию Apple и признали нарушение патентов данной компании в продуктах Samsung, в том числе патентов, связанных с первенством на дизайн и использованием мультитач (например, масштабирование щипком).

Размер компенсации, которую придется выплатить Samsung, определен в миллиард долларов. Ответные претензии Samsung отвергнуты. Решения о нарушении патентов и интеллектуальной собственности Apple затрагивают почти весь модельный ряд смартфонов и планшетов Samsung на базе платформы Android. На основании вынесенного решения компания Apple намерена добиться запрета на продажу смартфонов и планшетов Samsung в США. Слушания по этому делу пройдут 6 декабря 2012 года.

Комментируя вынесенное решение, компания Samsung указала на то, что результат разбирательства не следует рассматривать как победу Apple – это прежде всего проигрыш американских потребителей: вынесенное решение приведет к ограничению выбора, снижению инноваций и, возможно, к более высоким ценам. Вызывает сожаление, что манипулирование патентным правом может быть использо-

вано для предоставления одной компании монополии на оформление устройств в форме прямоугольника со скругленными углами и использование технологий, которые Samsung и другие компании совершенствовали каждый день. Вынесенное решение не является финальной точкой: противостояние с Apple ведется во многих судах по всему миру, и многие из судов уже признали несостоятельность претензий Apple.

**«Это прежде всего проигрыш американских потребителей.»**

Правозащитный ресурс Groklaw опубликовал свидетельства абсурдности схемы принятия решений в деле Apple против Samsung. Для рассмотрения присяжными было задано около 700 вопросов, по каждому из которых следовало вынести вердикт. Так как время на изучение материалов и принятия решений было ограничено, присяжные были лишены возможности детально проанализировать каждый эпизод и отделить существенные решения от второстепенных. В итоге ряд решений принял явно абсурдный характер: например, присяжные признали нарушение патента на дизайн iPhone в теле-

фоне с выдвинутой клавиатурой, четырьмя кнопками под экраном и огромным логотипом Samsung в верхней части (хотя суть дизайна Apple, напомним, как раз в отсутствии кнопок, а опасения – в возможности спутать аппараты Samsung с iPhone).

Google указывает, что большинство обвинений в нарушении патентов Apple не затрагивает непосредственно базовую платформу Android, а касается только дополнений, добавленных компанией Samsung поверх Android, например, домашней оболочки TouchWiz. Остальные патенты находятся на стадии пересмотра в патентном ведомстве США и, вероятно, в скором времени будут признаны несостоятельными. Что касается самого разбирательства, то Google считает, что все точки расставит процесс анализа состоятельности фигурирующих в деле патентов, который будет инициирован после подачи апелляции. По мнению Google, мобильная индустрия развивается очень быстро, и все нынешние производители мобильных систем, включая новых участников рынка, не создают ничего кардинально нового и строят свои разработки на идеях, известных уже не одно десятилетие. Google работает со своими партнерами над предоставлением потребителям инновационных и доступных по цене продуктов, и не желает, чтобы данное стремление пытались искусственно ограничить.

## «Корпорация добра» наносит ответный удар

21 августа 2012 года в окружном суде северного района Иллинойса Google в лице недавно купленной компании Motorola Mobility подала иск на корпорацию Apple, согласно которому Apple нарушает многочисленные патенты. Если суд признает факт нарушения патентов, итогом может стать запрет продаж в США практически всей линейки продуктов Apple, за исключением персональных компьютеров.

Под иск подпадают следующие устройства:

- » iPod Touch
- » iPhone 3GS, 4 и 4S
- » iPad2 и новый iPad (известный как iPad3)
- » Mac Pro и Mac Mini
- » iMac
- » MacBook Pro и MacBook Air

Иск требует удовлетворения в течение 15 месяцев с даты его подачи. Согласно иску, компании Apple должны запретить продажу устройств, использующих технологии радиосвязи, а также других портативных вычислительных устройств, устройств проигрывания музыки, компьютеров и всего остального, что подпадает под действия патентов.

Данные патенты, судя по всему, являются оригинальными изобретениями, не подпадающими под доктрину FRAND, однако некоторые из них имеют достаточно размытые формулировки, что усложнит компании Apple поиск путей защиты.

Пабло Чавес [Pablo Chavez], директор Google по связи с общественностью, выступил на конференции в Аспене с докладом, в котором раскры-

вал сегодняшнюю патентную систему и подчеркнул, что назрела необходимость искоренения практики выдачи патентов на программное обеспечение в США. В докладе сказано, что Google ставит под сомнение, что нынешнее положение с патентами на ПО способствует инновациям и отвечает потребностям потребителей.

По мнению Google, наблюдаемые в настоящее время патентные войны не приносят пользу потребителям, мешают развитию рынка и препятствуют развитию инноваций. При этом Чавес указал на то, что суть патентов на ПО существенно отличается от патентов в таких областях, как медицина, которые требуют длительных предварительных исследований и инвестиций.

ПЛАНШЕТ НА ВОЕННОЙ СЛУЖБЕ

# Высокие технологии – в армию

Министерство обороны России развивает специализированную сборку платформы Android.

**П**редставлен прототип первого российского планшетного компьютера, на базе ОС Android. Выпуск планируется к концу 2012 года. Основными пользователями должны стать сотрудники Минобороны. Операционная система обладает всеми функциями ОС Android, но не содержит негласных функций отправки личных данных пользователей в штаб-квартиру компании Google.

Компьютер получил название «РоМОС», как и операционная система (защищенная российская мобильная ОС). «Часть компонентов будут привозными, а сборка – российская на базе ЦНИИИЗСУ (головной институт Минобороны). Мы исключили из сборки Google-маркет по причине безопасности», отметил Андрей Стариковский, гендиректор научно-производственной компании на базе технопарка НИЯУ МИФИ. По его словам, для компьютера будет организован собственный магазин безопасных приложений.

Поставки планируется организовать в военное ведомство с возможностью применения компьютера в полевых условиях. Основные задачи планшета – шифрование данных, хранение криптоключей, карты местности, навигация ГЛОНАСС и GPS. «Будет два исполнения: штатное и военное (противоударное, водостойкое)», добавил г-н Стариковский.

➤ **Первый российский планшетный компьютер для нужд армии пока похож на все остальные планшеты.**



В отличие от предустановленных Android-прошивок, ОС РоМОС будет сертифицирована Министерством обороны для работы с конфиденциальной информацией, для чего будет проведен дополнитель-

ный аудит методов работы с внешними данными и удалена поддержка средств интеграции с сервисами Google.

Первый прототип новой ОС был продемонстрирован в середине сентября 2012 года на проходившей в Берлине выставке потребительской электроники IFA.

Доступный для потребителей вариант РоМОС также планируется выпустить до конца года. В частности, для широкой продажи будет подготовлен укомплектованный новой ОС планшет стоимостью 15 тыс. рублей. Позднее планируется подготовить тот самый водонепроницаемый и ударостойкий вариант планшета для военных. Поддержки 3G в планшете не будет.

## Немного истории

В сентябре 2010 года глава «Ростехнологий» Сергей Чемезов демонстрировал Дмитрию Медведеву, на тот момент президенту России, российский смартфон с поддержкой 4G под брендом Yota от оператора «Скартел», совладельцем которого является госкорпорация. «Это наш продукт – пока, к сожалению, выпускать мы его будем на Тайване, но в ближайшее время полностью перенесем производство в Россию», говорил тогда Чемезов. А гендирек-

тор «Скартела» Денис Свердлов пояснял, что этот телефон будет относиться к премиум-сегменту и по дизайну, и по начинке аппарата, и стоить будет соответственно – 25–30 тыс. руб. В продажу смартфон так и не поступил... В сентябре 2011 года Анатолий Чубайс представил Дмитрию Медведеву «уникальную» разработку – электронную книгу для школьников на базе технологии E-ink (электронных чернил), но и этот проект оказался провальным.

ВОТ ЭТО ДА!

# Мобильная ОС летит в Космос

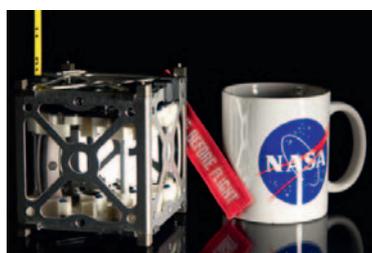
Телефоны с Android лягут в основу двух спутников NASA.

**В** NASA занялись любопытным экспериментом, цель которого – создание спутника из готовых деталей, а не разработанных силами агентства. Уже готовы две модели: PhoneSat 1.0 – на основе смартфона Nexus One и PhoneSat 2.0 – на основе Samsung Galaxy S.

Предполагается, что телефоны смогут какое-то время работать на орбите Земли сами по себе. Задача PhoneSat 1.0 – определить, на что способны стандартные системы телефона: радиопередатчики, гироскоп и камера, которая сделает несколько снимков Земли. К телефону

➤ **Крошечный спутник Phone Sat 1.0 будет снимать поверхность Земли «родной» камерой смартфона.**

подключат систему, которая его перезагрузит, если он перестанет подавать сигналы (она относится к профессиональному оборудованию, но тоже продается



свободно), а также внешний аккумулятор. В PhoneSat 2.0 войдут и «космические» разработки – радиопередатчик, мощности которого хватит для связи с Землей, солнечные батареи, приемник GPS и магнитная катушка, предназначенная для измерения магнитного поля Земли.

Главная фишка тут в том, что цена трех прототипов спутников не превысит 3500 долларов США – по меркам космических программ, это карманные деньги. Запуск спутников назначен на 2013 год и будет произведен в рамках программы Small Spacecraft Technology Program. LXF



## Android: Фото на карте



Наш эксперт

Джульетта Кемп пишет статьи для различных изданий и сейчас думает над тем, как бы воспользоваться HTTP-подключением своего телефона к серверу.

Джульетта Кемп пишет простую программу, которая привяжет фотографии к Google Maps с GPS.

На этом уроке мы рассмотрим интерфейс GPS, а также вернемся к некоторым другим компонентам системы, использованным в наших предыдущих статьях – API карты [Map], базы данных [Database] и камеры [Camera]. Подробно этот код мы разбирать не будем, но весь он присутст-

вует на DVD, поэтому при необходимости его можно скопировать, вставить и заставить работать (обратите внимание, что код, помещенный на DVD, не скомпилируется в том виде, как он есть – нужно создать новый проект, вручную или в *Eclipse*, и импортировать его).

### Шаг 1: Базовая настройка GPS

Интерфейс нашего приложения прост: тестовое поле (в котором будет показано местоположение после его определения) и три кнопки: Get Location [Определить местоположение], Take Photo [Сделать снимок] и Show Map of Photos [Показать карту с фотографиями]. На DVD найдите XML-файл `res/layout/main.xml`. Первые несколько методов в нашем *GPSPhoto* выглядят так:

```
private LocationManager lm; private TextView locationText;  
public void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);  
setContentView(R.layout.main);  
setUpButtons();  
locationText = (TextView) findViewById(R.id.location);  
lm = (LocationManager)  
getSystemService(LOCATION_SERVICE);  
}  
public void setUpButtons() {
```

### Информация о местоположении

Информация о местоположении включает не только широту и долготу, но и кучу других параметров. Для доступа к ним есть несколько методов в классе `Location`, а для их представления в человеко-читаемом виде пригодится метод `toString()` – результат получится примерно такой:

```
Location[mProvider=gps,mTime=1339624800000,  
mLatitude=55.0,mLongitude=0.0,mHasAltitude=true,  
mAltitude=0.0,mHasSpeed=false,mSpeed=0.0,  
mHasBearing=false,mBearing=0.0,mHasAccuracy=  
false,mAccuracy=0.0,mExtras=  
Bundle[mParcelledData.dataSize=4]]
```

Вот полная информация об этих параметрах:

- » **Provider** – провайдер местоположения, обычные значения – GPS или Network [Сеть].
- » **Time** – время съемки по Гринвичу (в миллисекундах с 1 января 1970 года).
- » **Latitude** и **Longitude** – как следует из названия, широта и долгота в градусах.
- » **hasAltitude, Altitude** – если привязка содержит информацию о высоте над уровнем моря (т.е. если

`hasAltitude` равно true), `Altitude` содержит высоту в метрах.

- » **hasSpeed, Speed** – если привязка содержит информацию о скорости, значение `Speed` будет ненулевым. По умолчанию информации о скорости не содержится.

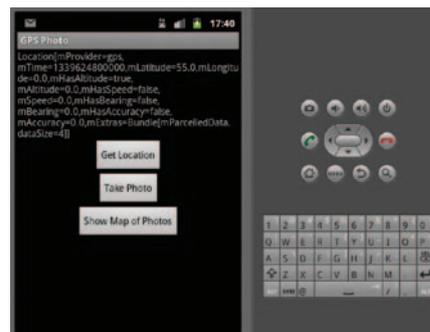
- » **hasBearing, Bearing** – если привязка содержит информацию об азимуте, `Bearing` содержит направление движения в градусах к востоку от географического севера. На эмуляторе этих данных нет, но на реальном устройстве они бывают.

- » **hasAccuracy, Accuracy** – если у провайдера есть данные о точности, `Accuracy` содержит значение допуска в метрах.

- » **Extras** содержит всю дополнительную информацию о привязке в виде пар «имя/значение». Одним из таких параметров может быть число спутников, использованное для получения привязки. Полная строка с параметрами расположения не слишком полезна программе, зато очень полезна программисту. Например, с помощью параметра

`Bearing` можно сориентировать стрелку на карте (также есть метод `bearingTo()`, который возвращает направление для заданной конечной точки).

Если имеется точная информация о высоте, то ее можно записывать через определенные интервалы времени, чтобы отслеживать свой подъем – это особенно удобно при езде на велосипеде.



```
Button getLocationButton = (Button) findViewById(
R.id.button_getLocation);
getLocationButton.setOnClickListener(new View.
OnClickListener() {
    public void onClick(View view) {
        getLocationButton();
    }
});
}
```

Задание объекта `LocationManager` в методе `onCreate()` позволит нам работать с различными провайдерами и параметрами местоположения. С его помощью мы получаем провайдер местоположения и текущее местоположение в методе `getCurrentLocation()`:

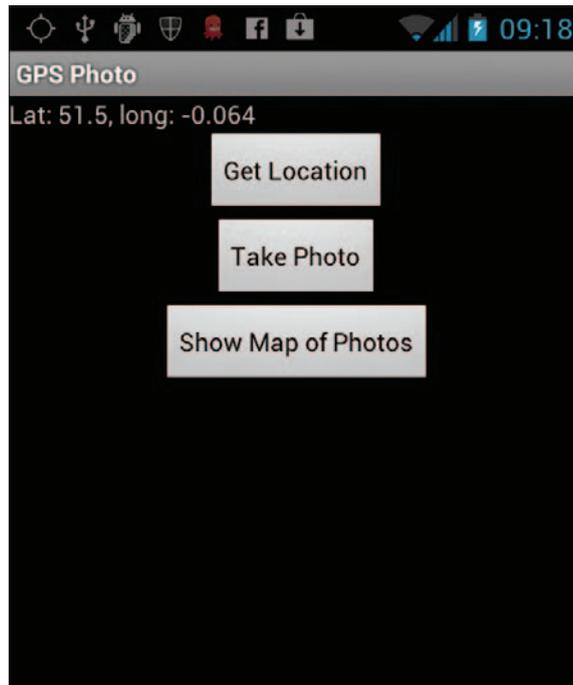
```
public void getCurrentLocation() {
    LocationListener locationListener = new LocationListener() {
        public void onLocationChanged(Location location) {
            locationText.setText(location.toString());
            lm.removeUpdates(this);
        }
        public void onStatusChanged(String provider, int status, Bundle extras) {}
        public void onProviderEnabled(String provider) {} public void onProviderDisabled(String provider) {}
    };
    lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, locationListener);
    Location lastKnownLocation = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
    if (lastKnownLocation != null) {
        locationText.setText(lastKnownLocation.toString());
    } else {
        locationText.setText("Последнее местоположение не получить");
    }
}
```

`LocationListener`, как можно догадаться по названию, отслеживает изменения местоположения и, заметив таковое, запускает метод `onLocationChanged()`. Когда местоположение меняется, мы устанавливаем значение текстового поля `[TextView] locationText` в строковое значение GPS. Остальные методы `LocationListener` несущественны – пока оставим их пустыми.

По созданию `LocationListener` его нужно зарегистрировать в `LocationManager` и выбрать провайдера. Здесь мы пользуемся провайдером GPS (если он недоступен, обработку ошибки см. далее). Чтобы воспользоваться провайдером GPS, также нужно установить следующие права доступа в **AndroidManifest.xml**:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Иногда на получение информации о местоположении требуется время – в процессе ожидания можно, если удастся, получить последнее известное местоположение вручную и использовать его, пока `LocationListener` не оповестит нас о следующем.



Если выполнить этот код, у нас получится результат следующего вида:

```
Location[mProvider=gps,mTime=1338588000000,
mLatitude=37.422005,mLongitude=-122.084095,
mHasAltitude=true,mAltitude=0.0,mHasSpeed=false,mSpeed=0.0,
mHasBearing=false,mBearing=0.0,mHasAccuracy=false,
mAccuracy=0.0,mExtras=Bundle[mParcelledData.dataSize=4]]
```

Здесь немало параметров, которые могут нам пригодиться – подробности см. во врезке «Информация о местоположении». Вытащим отсюда широту и долготу и выведем их с тремя знаками после запятой с помощью класса `DecimalFormat`. Для этого заменим вызов `setText()` таким кодом:

```
DecimalFormat latLngFormat = new DecimalFormat("#.###");
String lat = Double.toString(Double.valueOf(latLngFormat.
format(currentLocation.getLatitude())));
String lng = Double.toString(Double.valueOf(latLngFormat.
format(currentLocation.getLongitude()))); locationText.
setText("Шир: " + lat + ", долг: " + lng);
```

Отправьте одно значение перед перезагрузкой эмулятора, затем нажмите на кнопку `Get Location` и отправьте еще одно, и вы увидите, как значение изменится. Если вы сделаете это еще раз, значение меняться не будет, так как listener выключен. Конечно, можно оставить его включенным, но это довольно быстро разрядит батарею, поэтому лучше выключать и снова включать его при необходимости. В нашем случае – при каждом вызове `getCurrentLocation()`, т.е. при нажатии кнопки или при съемке фотографии.

» Три кнопки, а также широта и долгота моего текущего местоположения на реальном устройстве.

## Скорая помощь

Чтобы это заработало на эмуляторе, нужно запустить эмулятор, отправить ему местоположение с помощью `DDMS`, затем установить приложение.

## Шаг 2: Съемка фотографий

Для съемки фотографии можно вызвать стандартную программу работы с камерой с помощью намерения – **Intent**:

```
private Uri fileUri;
public void takePhoto() {
    Intent i = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    fileUri = getOutputMediaFileUri(MEDIA_TYPE_IMAGE);
```

```
i.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);
startActivityForResult(i, CAPTURE_IMAGE_ACTIVITY_REQ);
}
```

Найдите на DVD код метода `getOutputMediaFileUri()`, создающего уникальное имя файла, под которым будет сохранена фотография. Для исправления замеченной ошибки `fileUri` сделана

»

## Ошибка Занятия камеры при сохранении фотографии

В некоторых устройствах при возвращении результата Занятия камеры возникает ошибка, при которой URI сохраненной фотографии не возвращается (несмотря на то, что на самом деле фотография успешно сохранена).

Для обработки этой ошибки в методе `onActivityResult()` проверим, является ли возвращенное `Intent` пустым при коде результата `OK`. Если да, воспользуемся URI файла, который мы передали Занятию камеры.

Однако это значение нельзя просто сохранить в переменной класса, иначе оно будет потеряно при приостановке Занятия. Вместо этого воспользуемся методом `onSaveInstanceState()`:

```
private static final String SAVED_FILE_URI =
    "fileUri";
public void onSaveInstanceState(Bundle
    savedInstanceState) {
    if (fileUri != null) {
        savedInstanceState.putString(SAVED_FILE_URI,
            fileUri.toString());
    }
    super.onSaveInstanceState(savedInstanceState);
}
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (savedInstanceState != null) {
```

```
String fileString = savedInstanceState.
    getString(SAVED_FILE_URI);
    if (fileString != null) { fileUri = Uri.parse(fileString); }
}
// ....
}
```

Он сохраняет значение `fileUri` при приостановке Занятия и сохраняет его при возобновлении (если оно существует; мы проверяем, что состояние сохранено и не является пустым, чтобы избежать исключения `NullPointerException`).

Теперь все должно работать хорошо, несмотря на ошибку.

переменной класса – подробности во врезке «Ошибка Занятия камеры при сохранении фотографии».

Для хранения фотографий на внешней SD-карте нужно добавить следующее право доступа в **AndroidManifest.xml**:

```
<uses-permission android:name="android.permission.WRITE_
EXTERNAL_STORAGE" />
```

Получив `Intent` с местоположением фотографии, нужно кое-что с ним сделать:

```
protected void onActivityResult(int requestCode, int resultCode,
    Intent data) {
    if (requestCode == CAPTURE_IMAGE_ACTIVITY_REQ) {
        if (resultCode == RESULT_OK) {
            if (data == null) {
                // Здесь замечена ошибка! Фото надо сохранить в fileUri
                storePhoto(fileUri);
            } else {
                storePhoto(data.getData());
            }
        } else if (resultCode == RESULT_CANCELED) {
            // Пользователь отменил операцию; ничего не делаем
        } else {
            Log.e(TAG, "Вызов фото неудачен!");
        }
    }
}
```

Информацию об ошибке и борьбе с ней см. во врезке.

### База данных

Прежде чем писать метод `storePhoto()`, нужно создать базу данных фотографий: одну таблицу с тремя столбцами (URI фотографии, широта и долгота в формате с плавающей точкой). Для начала напишем вспомогательный класс базы данных, который использует API для работы с SQLite в Android:

```
public class GPSPhotoDatabaseHelper extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME
        = "GPSPhotoDatabase";
```

```
// Сюда пойдут константы для таблицы и шапки колонок
// А также константа для строки SQL; см. код на DVD
GPSPhotoDatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}
public void onCreate(SQLiteDatabase db) {
    db.execSQL(PHOTO_TABLE_CREATE);
}
public void onUpgrade(SQLiteDatabase db, int oldVersion,
    int newVersion) {
    db.execSQL(PHOTO_TABLE_DROP);
    onCreate(db);
}
}
```

Затем напишем метод `storePhoto()` в `GPSPhoto`, который будет вызываться из `onActivityResult()`:

```
private void storePhoto(Uri uri) {
    getLocation();
    GPSPhotoDatabaseHelper dbHelper =
        new GPSPhotoDatabaseHelper(getBaseContext());
    SQLiteDatabase db = dbHelper.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(GPSPhotoDatabaseHelper.KEY_FILE, uri.toString());
    values.put(GPSPhotoDatabaseHelper.KEY_GPS_LAT,
        getLocation.getLatitude());
    values.put(GPSPhotoDatabaseHelper.KEY_GPS_LNG,
        getLocation.getLongitude());
    db.insert(GPSPhotoDatabaseHelper.PHOTO_TABLE_NAME, null,
        values);
    db.close();
}
```

Этот код почти не требует пояснений: мы получаем текущее местоположение, переменную для работы с базой данных, затем с помощью переменной `ContentValues` записываем URI и местоположение снимка в базу данных. Теперь фотографии будут храниться в базе данных, и к ним будет легко обратиться при создании карты.

### Скорая помощь

Чтобы воспользоваться эмулятором камеры, запустите менеджер AVD и добавьте SD-карту в эмулируемое устройство.

## Шаг 3: Карты

Для настройки карт нужно собрать приложение с целью Google API (а не с обычной целью), и добавить следующую строку в файл **AndroidManifest.xml**:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Также в компоненте карты `[MapView]` нужно воспользоваться ключом API. Для отладки можно получить ключ, командой

```
keytool -list -alias androiddebugkey -keystore ~/.android/debug.keystore \
```

```
-keypass android -storepass android
```

и отправить получившуюся контрольную сумму по ссылке <https://developers.google.com/android/maps-api-signup>. Информация о том, как получить и инициализировать полноценный ключ, приведена на сайте Google. Затем укажите ключ в XML-файле:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout .... >
<com.google.android.maps.MapView
  android:id="@+id/mapView"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:enabled="true"
  android:clickable="true"
  android:apiKey="xxxxxxxxx"
/>
</RelativeLayout>
```

Чтобы воспользоваться этой XML-схемой MapView, создайте класс GPSPhotoMap:

```
public class GPSPhotoMap extends MapActivity {
// Override the methods
@Override
public void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.map);
}
@Override
protected boolean isRouteDisplayed() {
  return false;
}
}
```

Вызовите его из главного Занятия в методе **showMap()**, вызываемом при нажатии кнопки Show Map of Photos [Показать карту с фото]:

```
public void showMap() {
  Intent i = new Intent(getBaseContext(), GPSPhotoMap.class);
  startActivity(i);
}
```

Пока карта пуста; добавим же отметки, показывающие места, где сделаны наши снимки. Для этого воспользуемся оверлеем – и унаследуемся от `ItemizedOverlay`, так как нужно показывать несколько меток. Полный код (и предыдущие руководства) можно найти на DVD, а нам важны следующие методы:

```
public class PhotoLocationOverlay extends
ItemizedOverlay<OverlayItem> {
  private ArrayList<OverlayItem> photos = new
ArrayList<OverlayItem>();
  public PhotoLocationOverlay(Drawable marker) {
    super(boundCenterBottom(marker));
    populate();
  }
  // Добавим новый элемент на нашу карту
  public void addItem(GeoPoint p, String photoTitle, String
photoLocation) {
    OverlayItem photolItem = new OverlayItem(p, photoTitle,
photoLocation);
    photos.add(photolItem);
    populate();
  }
}
```

Элемент оверлея `OverlayItem` должен иметь три значения: точ- ко геолокации `GeoPoint`, заголовок и краткое описание `snippet-a`.



► Карта и отметка в GPSPhotoMap.

В методе **onCreate()** GPSPhotoMap добавим метод для прори- совки оверлея:

```
List<PhotoWithLocation> photoList;
public void onCreate(Bundle savedInstanceState) {
  // ...
  showPhotosOnMap();
}
private void showPhotosOnMap() {
  Drawable pin = this.getResources().getDrawable(R.drawable.pin);
  PhotoLocationOverlay photoOverlay = new
PhotoLocationOverlay(pin);
  MapView mapView = (MapView) findViewById(R.id.mapView);
  mapView.getOverlays().add(photoOverlay);
}
```

Объект `Drawable` – графическая метка, используемая для ка- ждого элемента; затем мы создаем и добавляем новый оверлей на карту, затем добавляем каждый элемент списка фотографий на карту.

## Объединим все вместе

Наконец, нам нужен метод, который получает список фотографий из нашей базы данных и передает его **showPhotosOnMap()**:

```
private void getPhotoList() {
  GPSPhotoDatabaseHelper db = new GPSPhotoDatabaseHelper
(getBaseContext());
  SQLiteQueryBuilder builder = new SQLiteQueryBuilder();
  builder.setTables(GPSPhotoDatabaseHelper.
PHOTO_TABLE_NAME);
  String[] PROJECTION = new String[] {
```

»

```

GPSPhotoDatabaseHelper.KEY_FILE,
GPSPhotoDatabaseHelper.KEY_GPS_LAT,
GPSPhotoDatabaseHelper.KEY_GPS_LNG
};
photoList = builder.query(db.getReadableDatabase(),
PROJECTION, null, null, null, null, null);
}
private void showPhotosOnMap() {
// ....
mapView.getOverlays().add(photoOverlay);
getPhotoList();
photoList.moveToFirst();
while (photoList.isAfterLast() == false) {
String uri = photoList.getString(photoList.getColumnIndex
(GPSPhotoDatabaseHelper.KEY_FILE));
float lat = photoList.getFloat(photoList.getColumnIndex
(GPSPhotoDatabaseHelper.KEY_GPS_LAT));
float lng = photoList.getFloat(photoList.getColumnIndex(GPSP
hotoDatabaseHelper.KEY_GPS_LNG));
GeoPoint gp = new GeoPoint((int)(lat * 1E6), (int)(lng * 1E6));
// Поле 'title' не используется, только GeoPoint and snippet
photoOverlay.addItem(gp, null, uri);
photoList.moveToNext();
}
photoList.close();
}

```

Для быстрого и легкого создания базы данных прекрасно подойдет SQLiteQueryBuilder. Достаточно указать только базу данных и проекцию (т.е. необходимые нам столбцы). Остальные параметры **query()** (здесь все они пусты) – выборка и аргументы выборки (т.е. какие столбцы нужно вернуть), выражения SQL GROUP BY и SQL HAVING, а также порядок сортировки. В проекции **PROJECTION** нам нужны три столбца, и мы забираем все фото из базы данных, порядок сортировки и все остальное не имеет значения. Поэтому мы выполняем запрос на ReadableDatabase с пустыми параметрами за исключением **PROJECTION**.

Функция возвращает Cursor, по которому можно проходиться в функции **showPhotosOnMap()**. Помните, что курсор нужно вручную перевести на первый элемент и затем продолжать итерации, пока он не дойдет до последнего. Каждый элемент добавляется в Overlay, а следовательно, добавится в Map View.

Скомпилируйте программу, и, сняв несколько фотографий и отправив несколько местоположений эмулятору, вы увидите на карте несколько отметок.

## Щелкаем по отметкам на карте

Теперь позаботимся, чтобы по щелчку на отметке показывалась сделанная фотография. Для начала создадим базовый метод обработки щелчка на метке с помощью диалога AlertDialog.

Добавьте следующие методы в PhotoLocationOverlay:

```

public PhotoLocationOverlay(Drawable marker,
Context context) {
super(boundCenterBottom(marker));
c = context;
populate();
}
protected boolean onTap(int index) {
OverlayItem item = photos.get(index);
AlertDialog.Builder dialog = new AlertDialog.Builder(c);
dialog.setTitle(item.getTitle());
dialog.setMessage(item.getSnippet());
dialog.show();
return true;
}

```

Затем измените данную строку метода **showPhotosOnMap()** в GPSPhotoMap, передав Context при создании оверлея:

```

PhotoLocationOverlay photoOverlay =
new PhotoLocationOverlay(pin, this)

```

Запустите программу, и вы увидите, что при щелчке по отметке теперь открывается окошко с путем к соответствующему файлу. Чтобы показать вместо него фотографию, нужен код несколько сложнее.

Употребим обычный Dialog вместо AlertDialog. Перепишем метод **onTap()**:

```

protected boolean onTap(int index) {
OverlayItem item = photos.get(index);
Dialog dialog = new Dialog(c);
LayoutInflater inflater = (LayoutInflater)
c.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
LinearLayout photoAlert = (LinearLayout) inflater.inflate
(R.layout.photoalert, null);
ImageView photoImage = (ImageView) photoAlert.
findViewById(R.id.photo);
try {
InputStream in = new java.net.URL(item.getSnippet()).
openStream();
Bitmap bm = BitmapFactory.decodeStream
(new FlushedInputStream(in));
photoImage.setImageBitmap(bm);
// Обрабатываем все возможные исключения
} catch (MalformedURLException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
dialog.setContentView(photoAlert);
dialog.show();
return true;
}

```

Чтобы показать в Dialog изображение вместо текста, нужно создать и использовать пользовательскую раскладку. LayoutInflater ее получит, а мы сможем получить ссылку на ImageView. Ваш XML-файл **layout/photoalert.xml** должен выглядеть так:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/
res/android"
android:layout_width="wrap_content"
android:layout_height="wrap_content" >

```



➤ Из карты выросли деревья и река. Чтобы стало еще лучше, доработайте код – пускай фотография повернется на нужный угол.

```
<ImageView
    android:id="@+id/photo"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Наконец, нам требуется преобразовать фотографию в формат bitmap и поместить ее в ImageView. В идеальном варианте нам следовало бы вызвать **BitmapFactory.decodeFile(file)**.

К сожалению, замечена неприятная ошибка, способная привести к тому, что файлы JPG (сохраненные камерой аппарата) могут

не декодироваться (см. <http://code.google.com/p/android/issues/detail?id=6066>).

Вам нужно переписать класс **FlushedInputStream**; код этого статического класса помещен на DVD. Создав bitmap, покажите окно, и все готово.

Чтобы окно закрывалось без нажатия кнопки Back [Назад], добавьте в код всего одну строку:

```
dialog.setCanceledOnTouchOutside(true);
```

Теперь при щелчке на любом свободном месте окно закроется, и мы вернемся к карте.

## Шаг 4: Улучшаем GPS

Чтобы определить местоположение более надежно, перепишем метод **isBetterLocation()**:

```
protected boolean isBetterLocation(Location newLocation,
    Location oldLocation) {
    if (oldLocation == null) {
        return true;
    }
    long timeDiff = newLocation.getTime() - oldLocation.getTime();
    if (timeDiff > TWO_MINUTES) {
        return true;
    } else if (timeDiff < -TWO_MINUTES) {
        return false;
    }
    int accuracyDiff = (int) (newLocation.getAccuracy() -
        oldLocation.getAccuracy());
    if (accuracyDiff < 0) { return true; }
    else if (accuracyDiff > 0) { return false; }
    return false;
}
```

Он проверяет три вещи:

- 1 Если текущего местоположения нет, то новое всяко будет лучше, поэтому возвращаем **true**.
- 2 Если разница во времени между текущим и новым местоположением больше двух минут, то пользователь, вероятно, переместился в пространстве, поэтому воспользуемся новым местоположением (возвращаем **true**).
- 3 Точность выражается в метрах (см. врезку «Информация о местоположении»). Если точность текущего местоположения меньше, чем нового, то (**new\_accuracy — old\_accuracy > 0**), и текущее местоположение лучше (возвращаем **false**). Если наоборот, то лучше новое местоположение, и мы возвращаем **true**. Если разницы нет, сохраняем текущее местоположение.

Теперь можно вызывать эту функцию в обоих случаях получения нового местоположения – при его обновлении и при получении последнего известного местоположения. На практике мы выделим его в отдельный метод и вызовем его:

```
public void getCurrentLocation() {
    LocationListener locationListener = new LocationListener() {
        public void onLocationChanged(Location location) {
            checkAndSetLocation(location);
            lm.removeUpdates(this);
        }
    };
    // ...
    Location lastKnownLocation = lm.getLastKnownLocation(
        PROVIDER);
    if (lastKnownLocation != null) {
        checkAndSetLocation(lastKnownLocation);
    }
}
```

```
}
private void checkAndSetLocation(Location location) {
    if (isBetterLocation(location, currentLocation)) {
        currentLocation = location;
    }
    DecimalFormat latLngFormat = new DecimalFormat("#.###");
    String lat = Double.toString(
        Double.valueOf(latLngFormat.format(currentLocation.
            getLatitude())));
    String lng = Double.toString(
        Double.valueOf(latLngFormat.format(currentLocation.
            getLongitude())));
    locationText.setText("Lat: " + lat + ", long: " + lng);
}
```

Мы также проверим, доступен ли провайдер GPS, и если нет, воспользуемся провайдером сетевой мобильной связи, добавив следующие строки в **getCurrentLocation()**:

```
if (!lm.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
    PROVIDER = LocationManager.NETWORK_PROVIDER;
} else {
    Toast.makeText(this, "GPS недоступно; берем сетевую
    ячейку",
        Toast.LENGTH_SHORT).show();
    PROVIDER = LocationManager.NETWORK_PROVIDER;
}
lm.requestLocationUpdates(PROVIDER, 0, 0, locationListener);
```

Можно и еще улучшить код, написав методы **onStatusChanged()**, **onProviderEnabled()** и **onProviderDisabled()** в **LocationListener**. Например, можно выдавать окошко с оповещением при отключении провайдера GPS (сначала проверяется, что текущим провайдером является GPS, затем выдается окошко), чтобы пользователь включил его снова.

Если вы хотите пользоваться только провайдером GPS (не сетевым), в этот момент можно даже закрыть приложение, но учтите, что это ограничивает функциональность программы и может очень досажать пользователем, которые предпочитают не оставлять GPS включенным и довольны примерным местоположением.

В методе **onStatusChanged()** можно также выдать окошко, если новый статус равен **LocationProvider.OUT\_OF\_SERVICE**, **TEMPORARILY\_UNAVAILABLE** или **AVAILABLE**. Поработайте с этими методами и выберите лучшие варианты для своего приложения.

Желая еще улучшить программу, можно добавить кнопку для отправки фотографии и местоположения в Twitter или в блог.

Также можно улучшить обработку нескольких фотографий, снятых в одном и том же или близких местах (например, показав список при щелчке по отметке). Как всегда, добавляемые возможности и варианты использования фреймворка и API остаются на ваше усмотрение. **LXF**

# Сравнение

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

## Свободные лицензии

Все программы с открытым кодом свободны, но некоторые свободнее других. Бен Эверард сравнивает лицензии, дающие нашим программам свободу.



### Про наш тест...

Поскольку на самом деле мы тестируем не программы, руки приложить нам было не к чему. Но мы сравнивали, как эти лицензии защищают свободу пользователя. Мы решили использовать определение свободы, заключенное в перечислении четырех свобод от Free Software Foundation (FSF). Это:

- »» Свобода использовать программу для любых целей.
- »» Свобода изучать то, как работает программа, и изменять ее, чтобы она выполняла поставленную вами задачу по вашему усмотрению. Непременным условием для этого является доступ к исходному коду.
- »» Свобода распространять копии, чтобы помочь ближнему.
- »» Свобода распространять модифицированные вами версии. Таким образом вы позволите всему сообществу извлечь пользу из внесенных вами изменений. Непременным условием для этого является доступ к исходному коду.

### Наша подборка

- » Apache
- » BSD-style
- » GPLv2
- » GPLv3
- » LGPL

**М**ы часто говорим о программах с открытым кодом так, словно это — одна-единственная идея, и одна-единственная философия. На самом деле в этой области имеется множество течений и движений, и каждое определяется собственной лицензией. Эти документы содержат в себе идеалы группы кодеров точно так же, как политический манифест содержит в себе идеалы политической партии; единственное отличие в том, что с юридической точки зрения они обязательны к исполнению.

Они описывают права, которые разработчики передают пользователям, и требования, которые они предъявляют будущим кодерам.

Единого и унифицированного определения открытого кода в природе не существует, и ряд проектов имеет собственный список утвержденных лицензий. Самые известные из этих списков поддерживаются Open Source Initiative, Debian и Free Software Foundation. Различия между ними незначительны, но лицензии здесь будут рассматриваться по всем спискам.

Мы решили рассмотреть пять лицензий, которые охватывают и весь спектр философий открытого кода, и большинство имеющихся программ с открытым кодом. Это (в скобках указаны самые известные программы): GPLv2 (ядро Linux), GPLv3 (разделы GNU большинства дистрибутивов Linux, такие, как *Bash* и *GCC*), Apache (web-сервер *Apache*, Android за исключением ядра, недавние версии *OpenOffice.org*), LGPL (*LibreOffice*) и BSD (все операционные системы BSD, *X.org*). Читайте, чтобы узнать, как они выглядят под нашим микроскопом.

# Copyleft

## Создание кода, который нельзя закрыть.

**В**ероятно, самое сильное расхождение наблюдается между теми лицензиями, которые требуют, чтобы все модификации были с открытым кодом, и теми, которые этого не требуют. В первых (известных под названием copyleft – потому что это слово противоположно по значению copyright, уловили?) любые разработчики, которые добавили какие-либо функции к программе, должны поделиться этими функциями с другими пользователями. В последних они оставляют эти функции исключительно для себя (или требуют за них плату).

Первой основной лицензией, которая ввела copyleft и по-прежнему остается самой популярной, является GPL. Это есть во всех формах GPL.

От разработчиков требуется при распространении своих программ предоставлять исходный код. Это означает, что они могут разрабатывать код в частном порядке и делиться им с общественностью только тогда, когда код их устроит. Конечно, многие проекты предпочитают делиться своим кодом раньше. Если они оставляют свои модифицированные программы себе, им не придется делиться изменениями с другими пользователями. Это слегка

нарушается в среде облачных вычислений, поскольку здесь разработчики не распространяют программ, а выдают другим пользователям доступ к ним. В этой ситуации разработчик может изменить программу GPL и продать пользователю доступ к ней, но не возвращать свои изменения в сообщество. Для борьбы с этим FSF выпустили AGPL (Affero General Public License), которая требует, чтобы разработчики делились своим кодом, если они открывают приложение для общественности, вне зависимости от того, распространяют ли они эту программу.

Ни лицензия Apache, ни BSD не имеют никакой формы copyleft, и компании имеют право – и используют его – включать свободный код в коммерческие продукты. Сторонники этого типа лицензии заявляют, что в таком виде она более дружелюбна к бизнесу, и по этой причине, в конечном итоге, оказывается в большем количестве продуктов.

Например, когда Microsoft решил включить сетевой протокол TCP/IP в Windows, они взяли немалую часть кода из BSD. И это означало, что Microsoft получил рабочий сетевой протокол, взаимодействующий с большинством других систем.



» «Хорошие художники копируют, великие – воруют» – Никае Бен Зверд. И некоторые лицензии это допускают.

Возможно, свободное ПО от этого не выиграло, но зато выиграли компьютерные технологии в целом, ведь это привело к тому, что большинство компьютеров (с Windows) стали работать лучше. Подобным же образом Apple решил подвергнуть радикальному редизайну свою ОС, и построил ее на коде FreeBSD. Это означало, что большая часть работы его дизайнеров была потрачена на создание превосходного интерфейса, и им не надо было переживать по поводу ядра. Цель лицензий с copyleft – улучшение мира свободного ПО, а лицензий без него (иногда именуемых разрешительными) – улучшение мира компьютерных технологий в целом.

Многие из наиболее широко используемых программ с открытым кодом выходят под разрешительными лицензиями – скажем, Android (кроме ядра), Chromium, веб-сервер Apache и OpenOffice.org. Ядро Linux является одной из немногих важных программ под лицензией copyleft.

### Вердикт

GPLv2  
★★★★★  
GPLv3  
★★★★★  
LGPL  
★★★★★  
Apache  
★★★★★

BSD-style  
★★★★★

» Категория типа «все или ничего», где высший балл получает FSF.

# Тивоизация

## Изменение «внутренностей».

**К**омпьютерный мир славится своим обескураживающим языком, но даже в нем слово «tivo-изация» является в некоторой степени архаичным. Tivo – это устройство, позволяющее пользователям смотреть и записывать цифровое ТВ, и оно содержит код под лицензией GPL. Производитель, в соответствии с лицензией, открыл свой код для общественности. Зато само оборудование было закрыто, и поэтому пользователи не могли модифицировать код и вносить свои об-

новления в оборудование. И это привело к расколу в сообществе открытого кода.

Линус Торвалдс поддержал право компании поступать таким образом – раз уж они открыли код для общественности, другие производители оборудования могли воспользоваться его модификациями и улучшениями, внесенными в него. А вот Ричард Столлмен счел, что пользователи должны иметь не только возможность получить доступ к коду, но и возможность изменять его на самом устройстве.

Пункт, препятствующий тивоизации, содержат только лицензии LGPLv3 и GPLv3: обе требуют предоставить пользователю возможность устанавливать собственные версии программы на оригинальное устройство. Однако они допускают исключения для устройств, которые физически не могут изменять ПО – например, если ПО занесено в ПЗУ, которое невозможно переписать.

Именно этот пункт способствовал тому, что Ubuntu выбрал программу загрузки под лицензией GPLv2 для будущих версий, которые содержат безопасную загрузку. Однако FSF заявил, что это произошло из-за непонимания лицензии в Canonical. Итак, пока этот пункт присутствует, продолжают споры о том, что же он означает. Так же, как и с copyleft, это совсем не означает слабости других лицензий – нет, это выбор, позволяющий дать разрешение.

### Вердикт

GPLv3  
★★★★★  
LGPL  
★★★★★  
Apache  
★★★★★  
BSD-style  
★★★★★  
GPLv2  
★★★★★

» До этого есть дело не всем, но если есть, выбирайте лицензию точно.



» Перед установкой модулей расширения убедитесь, что они совместимы с вашей версией CMS.

# Индивидуальные моменты

Дополнительные пункты, о которых вам надо знать.

**К**аждая из лицензий имеет свой пункт. Это особые требования, предъявляемые ими к коду и не попадающие ни в одну из имеющихся здесь категорий. А это оказывается крайне неудобным, когда для нескольких разных лицензий (например, вариаций BSD) используется общий «накрывающий [umbrella]» термин.

Об этом важно знать, потому что неосведомленным кодерам, которые даже не подозревают, что им придется подчиняться лиш-

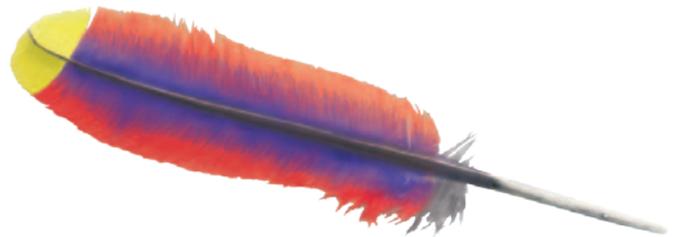
ним пунктам – в основном менее известным, чем те, что мы обсудили – легко угодить в ловушку. Здесь мы рассмотрим такие вещи, чтобы вы принимали их в расчет при разработке собственных программ.

Мы также рассмотрим разные версии каждой лицензии и их дрейф с течением времени. Как вы сами убедитесь, для одних лицензий это проблема в большей степени, а для других – в меньшей.

## Apache ★★★★★

Лицензия Apache радует прямоотой – в том смысле, что имеет только две версии, изобретательно названные «один» и «два». Вторая версия вышла в 2004 году, и почти все программы перешли на нее, так что мы рассмотрим только эту версию.

Если программа сопровождается текстовым файлом NOTICE, этот файл должен быть включен в модифицированные версии (возможно, в форме, просматриваемой из программы). В этот файл можно добавлять дополнительную информацию, при условии, что она не перепутается с уже имеющимся текстом. Информация о лицензии должна присутствовать в каждом файле, и если вы что-то меняете, изменения должны быть задокументированы в файле. Помимо этих двух ограничений и патентной информации, включаемой повсеместно, вы более или менее вольны делать с кодом под лицензией Apache все, что вам заблагорассудится.



## BSD ★★★☆☆

Лицензии BSD – самые простые, и практически не налагают никаких обязательств на тех, кто модифицирует код. Однако из-за их простоты очень многие их переписывали, и имеется множество разных версий.

Изначально в лицензии было четыре пункта, которые можно изложить так:

- » Дистрибутив в исходной форме должен включать уведомление об авторских правах.
- » Дистрибутив в двоичной форме должен включать уведомление об авторских правах в документации или файлах помощи.
- » В рекламном материале должны упоминаться разработчики.
- » Имена разработчиков неприменимы для рекламы производных продуктов.

Третий пункт не согласовывался с GPL и был удален в 1993 году, когда вышла новая лицензия BSD, уже из трех пунктов. Команда FreeBSD еще сильнее упростила лицензию, удалив из нее пункт четыре.

# Читаемость

Потому что мы любим понимать, что написано мелким шрифтом.

**П**одавляющее большинство пользователей программ с открытым кодом – и вообще программ – не являются юристами. Они не разбираются в тонкостях лицензий на ПО. Дело еще усложняется тем, что программы ныне буквально в считанные часы распространяются по всему земному шару, и по этой причине лицензия должна соответствовать сотням разных законодательств. У лицензий BSD здесь явное преимущество: они значительно проще остальных.

В Apache немного больше текста, и она звучит более обоснованно с точки зрения права, но отлично понятна и не юристам, благодаря своим простым требованиям.

Несмотря на большую сложность, GPLv2 остается весьма понятной и читаемой. А LGPL по большей части – то же самое, что GPL для версии 2.1.

Чего не скажешь о версии 3 этих лицензий. В определенной степени это связано с попытками учесть множество вариантов случаев, которые FSF сочла непод-

обающимися, вместо того, чтобы дать общее определение. Мы предоставим заключительное слово по этой теме разработчику SelectTor:

«Вы можете также спросить, почему GPLv2, а не GPLv3? Это просто: я тупой, и через GPLv3 продраться не смог. По моему мнению, лицензия GPLv2 – куда лучше написанная лицензия для простого человека, она отвечает моим потребностям, и мне глубоко наплевать, сколько юристов они там привлекли к созданию GPLv3».

## Вердикт

- GPLv2 ★★★★★
- Apache ★★★★★
- BSD-style ★★★★★
- LGPL ★★★★★
- GPLv3 ★★★★★

» Я не юрист, однако желаю понимать, что гласит моя лицензия.



### GPLv3 ★★★★★

В GPLv3 намного больше пунктов, чем в других наших лицензиях. Часть из них уже рассматривалась в этой статье. Не рассматривалась пока что пункт о технических средствах защиты авторских прав – Digital Rights Management (DRM). Это когда компания продает вам цифровой продукт (например, видео или музыку), но шифрует его так, чтобы он мог использоваться только на одном устройстве.

По этому поводу GPLv3 говорит: «Если вы передаете защищенное произведение, вы отказываетесь от юридического права запрещать обход техническими средствами до такой степени, когда подобный обход...» Иными словами, вы можете попытаться вставить DRM в код под GPLv3, но не можете помешать другим найти средства и возможности его обойти. А поскольку люди, пытающиеся его обойти, будут иметь доступ к вашему исходному коду, велика вероятность того, что они преуспеют в своих намерениях.



### GPLv2 ★★★★★

Версия 2 GPL – это замечательно четкая лицензия, чем, вероятно, и объясняется ее популярность. Помимо copyleft, она налагает очень мало ограничений на то, как вам допускается использовать код. Фактически, единственным присутствующим в ней ограничением – не разрешить налагать другие ограничения! Например, вам не разрешено ограничивать области применения программного продукта.

В прошлом этот аспект вызвал разногласия, поскольку некоторые проекты желали обеспечить неприменимость разработанного ими кода для военных целей.

GPU, клиент Gnutella, создал на основе GPL новую лицензию, изменив этот пункт. Согласно этой лицензии, код можно было использовать только в мирных целях. Это нарушило совместимость с оригинальной лицензией GPL, и лицензия не получила популярности в других проектах.



**Free as in Freedom**

### LGPL ★★★★★

LGPL выходит в нескольких формах. Самая известная – 2.1, которая примерно соответствует GPLv2, и версия 3 – созвучная GPLv3. И хотя во многих аспектах они похожи, их отличие лежит в их совместимости. Вероятно, самое странное в LGPL – то, что люди, ее написавшие (FSF), не одобряют ее использования.

Они согласны с тем, что для нее есть свое место (иначе они бы ее не создали), но при этом считают, что ее следует применять только в особых обстоятельствах. Ее разработали, чтобы создать библиотеку C под свободной лицензией, ссылку на которую могла содержать любая коммерческая программа. По их мнению, эту лицензию стоит применять исключительно ради внедрения свободного ПО там, где в ином случае будет использована коммерческая программа. Для вящей путаницы, буква L означала разные вещи: изначально – «библиотечная [library]», а сейчас – «умаленная [lesser]».

## Совместимость

Хорошо ли лицензии уживаются друг с другом?

**О**бластей, в которых встает проблема совместимости, две: конвертирование кода из одной лицензии в другую и связывание.

Связывание – это когда код в программе ссылается на библиотеку. GPLv2 и дальнейшие версии запрещают внешнему коду под несвободной лицензией ссылаться на код под лицензией GPL. За это ее иногда называют вирусной лицензией – она пытается «заразить» свободой все программы в системе. FSF ведет спи-

сок лицензий, которые, по его мнению, совместимы, на сайте [www.gnu.org/licenses/license-list.html](http://www.gnu.org/licenses/license-list.html). Но FSF обнаружил, что «вирусная» природа тормозит переход на свободное ПО. Людей заботит ее агрессивность, и они предпочитают библиотеки, не так жестко лицензированные, обычно не copyleft. Чтобы с этим бороться, FSF выпустил LGPL (Lesser или Library GPL). Код, выходящий под этой лицензией, имеет тот же copyleft в GPL, но может быть связан с кодом, не являющимся свободным.

Лицензии BSD в общем наиболее совместимы, и код, выходящий под ними, может быть связан с ними или перейти на другие, менее разрешительные лицензии. Исключение – лицензия BSD из четырех пунктов (старая). Пункт о рекламе лишает ее совместимости с GPL.

Так как под лицензией Apache вышел ряд важных программ, FSF обеспечил ее совместимость с версией 3 GPL, несовместимой с более ранними версиями. Она ограничена лицензией Apache версии 2.

### Вердикт

BSD-style

★★★★★

LGPL

★★★★★

Apache

★★★★★

GPLv2

★★★★★

GPLv3

★★★★★

» Будьте осторожнее, объединяя коды под разными лицензиями.

# Патенты

Не только copyright сохраняет закрытость кода.

**В** последние годы патентное законодательство стало полем боя, где ведут сражения богатейшие компании мира. Патенты защищают скорее идеи, а не реализацию. В качестве недавнего примера – Apple inc. создали функцию Slide to unlock, чтобы разблокировать функции на устройствах iOS. Закон об авторских правах не позволяет другим компаниям напрямую копировать использованный код и создавать его копии.

Однако Apple хотел защитить саму идею, а не просто код, использованный для его создания. Он подготовил соответствующую заявку и получил патент на идею Slide to Unlock, запрещающий другим производителям реализовывать данную функцию, независимо от того, будут ли они напрямую копировать код (примечание: этот патент может быть признан недействительным. На момент написания статьи он рассматривался в суде).

Это означает, что софтверные лицензии, созданные для защиты от copyright, не могут служить адекватной защитой от патентной угрозы.

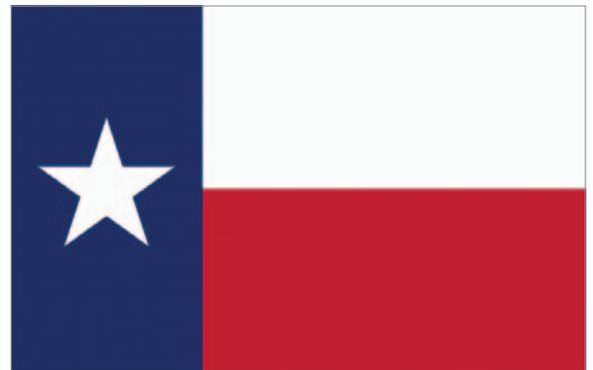
Лицензии BSD не предлагают никакой защиты в этой области. Софтверная компания может выпускать под одной из этих лицензий код, и при этом требовать уплаты патентных пошлин от тех, кто использует эти программы.

LGPL в этой области предлагает такую же защиту, как и соответствующая версия GPL. Пункт о патентах GPLv2 часто именуют «Свобода или смерть» – что всего лишь означает следующее: если вы ограничиваете права пользователей, вы теряете право использовать программу. Одной из главных причин обновления GPL до версии три было желание сделать раздел защиты от патентов более недвусмысленным.

В новой версии лицензии эта защита укрепилась, в ответ на разбирательство между Novell и Microsoft. При этом Novell заплатил Microsoft, а Microsoft, в свою очередь, согласился не предъявлять иск клиентам Novell.

Все это допускалось лицензией GPLv2, поэтому FSF ввел в GPLv3 пункт, который устанавливал, что если вы предъявляете подобный иск, он будет относиться ко всем, кто использует программу, а не только к тем, кто получил ее от вас.

GPLv3 предлагает превосходную защиту от патентов, но редко применяется в конкурентных областях. Возможно, выдающейся лицензией в этой сфере является Apache – не по юридическим причинам, а по причине широкого распространения в самой конфликтной технологической области, мобильных телефонах. Стычки Apple с разнообразными производителями



» Большинство исков по патентным вопросам прошло в Техасе, где применяется весьма ограничительное толкование патентного законодательства.

телефонов с Android всем известны, а вот между конкурирующими производителями телефонов на Android судебных разборок нет, хотя у каждого из них внушительное патентное портфолио. Отчасти за это нужно благодарить раздел 3 лицензии Apache, где просто говорится, что если вы распространяете программу под этой лицензией, вы должны передать и патентные права на нее.

Примечательно, что она не вводит такой же пункт, как GPLv3, и было несколько патентных соглашений между производителями Android и Microsoft, невозможных под лицензией FSF.

## Вердикт

GPLv3 ★★★★★  
 LGPL ★★★★★  
 Apache ★★★★★  
 GPLv2 ★★★★★  
 BSD-style ★★★★★

» Большинство лицензий на СПО справились с патентной угрозой.

# Количество программ

Лицензия хороша так, как хороши программы под ней.

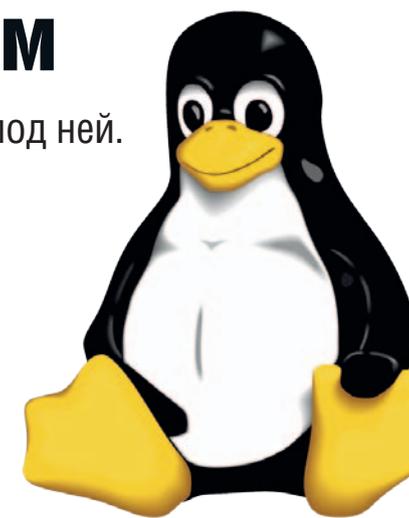
**Д**аже отлично защищающая пользователей лицензия бесполезна, если никому не нужны программы под ней. Итак, следует рассмотреть, что именно лицензии защищают. Black Duck Software ведет на <http://osrc.blackducksoftware.com/data/licenses> список программ с открытым кодом, и здесь мы можем увидеть, насколько широко используется каждая из них. Десятка самых популярных лицензий выглядит так [на момент написания статьи – прим. ред.]:

- 1 – GPL v2 (36 %)
- 2 – MIT Licence (12 %)
- 3 – Apache Licence (10 %)
- 4 – GPL v3 (10 %)
- 5 – BSD Licence (7 %)
- 6 – Artistic Licence (6 %)
- 7 – LGPL v2.1 (6 %)

- 8 – LGPL v3 (2 %)
- 9 – Code Project Open Licence (2 %)
- 10 – Microsoft Public Licence (1 %)

В этом Сравнении мы объединили лицензии MIT и BSD, поскольку они одинаковы. Конечно, общеизвестно высказывание, что есть ложь, есть наглая ложь, а есть статистика. Этот список не пытается сравнить ни качество программ, выходящих под каждой лицензией, ни широту их распространения, и, возможно, вовсе не характеризует их использование.

Но этот список в основном отражает наш опыт. Возможно, самое удивительное в нем то, насколько GPLv2 до сих пор популярнее v3. На то есть несколько причин. Отчасти это объясняется тем, что более старые проекты не могут перевести код на новую лицензию по юридическим при-



» Под GPLv2 выходит больше всего программ – и наше любимое пингвиное ядро.

чинам. Однако многие разработчики просто предпочитают старую версию, и полагают, что v3 отнимает больше свобод, чем предоставляет.

## Вердикт

GPLv2 ★★★★★  
 Apache ★★★★★  
 BSD-style ★★★★★  
 GPLv3 ★★★★★  
 LGPL ★★★★★

» Иногда все это сводится к простому конкурсу на популярность.

## Лицензии на свободное ПО

**Вердикт**

**М**ы тестировали лицензии на соответствие определению свободного ПО от FSF, и неудивительно, что победила та лицензия, которую именно FSF и предпочитает.

Однако это совсем не обязательно означает, что она лучшая. Каждая лицензия открытого кода должна найти баланс между тем, что она разрешает, и тем, что она ограничивает, во имя свободы пользователя. Факт нечастого применения GPLv3 за пределами проекта GNU показывает, что она налагает текущие ограничения во имя будущих свобод намного жестче, чем это устраивает разработчиков.

Более чистое определение свободы может рассматривать такие ограничения, как copyleft, в качестве дополнительной реализации основной свободы делать с кодом все, что угодно. Это состояние особенно хорошо выражено в BSD-подобной лицензии DWTFYWWI [Делайте Все, Что Заблагорассудится], всего с одним условием:

**0** Автор гарантирует всем разрешение делать все, что угодно, какими бы, черт побери, ни были последствия.

Это весьма похоже на то, что обычно называют общественным доменом. Другие лицензии BSD обычно добавляют несколько ограничений, обычно требующих не забыть упомянуть авторов.

Две вышеперечисленных лицензии являются собой крайности философии открытого кода. Между ними лежит обширная область моральных основ, где и обитает большинство разработчиков.

Например, Линус Торвалдс громогласно поддерживал GPLv2, и не намерен переводить свое ядро на новую версию лицензии. Многие согласны с Линусом, и GPLv2 остается лицензией номер один согласно статистике Open Source Resource Centre. Вряд ли ситуация изменится в ближайшем будущем, потому что она выражает

мысли людей об открытом коде: copyleft, но с ограничениями. Лицензия Apache, похоже, сохранит доминирование на корпоративной стороне открытого кода. Многие костюмоносные крохоборы боятся copyleft, что может препятствовать присоединению компаний к движению открытого кода. Благодаря лицензии Apache подобные опасения не являются проблемой, а возможность патентных угроз минимизирована.



» Open Source Initiative ведет список из 69 лицензий открытого кода.

**«Между ними — обширная область моральных основ, где обитает большинство.»**

**I****GPLv3**

Сайт: [www.gnu.org/licenses/gpl.html](http://www.gnu.org/licenses/gpl.html)

» Рьяно защищает свободу пользователя, но трудна для понимания.

**IV****Apache**

Сайт: [www.apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0)

» Дружественная к бизнесу опция, защищающая от патентных войн.

**II****GPLv2**

Сайт: [www.gnu.org/licenses/gpl-2.0.html](http://www.gnu.org/licenses/gpl-2.0.html)

» Просто содержит определение copyleft, и охватывает массу программ.

**V****LGPL**

Сайт: [www.gnu.org/licenses/lgpl.html](http://www.gnu.org/licenses/lgpl.html)

» Иногда бывает полезна, но могут обнаружиться опции и получше.

**III****BSD-style**

Сайт: [en.wikipedia.org/wiki/BSD\\_licenses](http://en.wikipedia.org/wiki/BSD_licenses)

» Истинная свобода, но несколько опорочена обилием версий.

**Обратная связь**

Вы пользовались какой-либо из этих систем? Считаете их потрясающими? Расскажите нам: [ixf.letters@futurenet.co.uk](mailto:ixf.letters@futurenet.co.uk)

**Рассмотрите также...**

Пригодных лицензий существует много – некоторые считают, что слишком много, и что наличие практически идентичных лицензий только путает пользователей. Помня об этом, лучше использовать одну из вышеперечисленных, а не менее известную лицензию, если только у вас нет на то особых причин. Однако есть еще две лицензии, которые применяются в крупных проектах: Mozilla Public Licence и Artistic Licence (Perl).

В основном они уместны со своими родительскими проектами.

Если вам нужна лицензия не для кода, есть смысл взглянуть на лицензии Creative Commons. В них есть ряд положений, которые можно смешивать и сочетать в соответствии с вашей философией. Основная лицензия – Attribution – требует только того, чтобы те, кто использует вашу работу, не забывали вас упоминать. К ней можно добавить

ShareAlike (примерно то же, что copyleft), NonCommercial и NoDerivs (которая означает, что вашу работу можно распространять только в неизменном виде). Выбирайте ту их комбинацию, которая удовлетворит ваш вкус. Лицензии Creative Commons сделаны простыми для понимания, и в них есть значки, которые можно добавлять к визуальным работам, без необходимости добавлять весь текст лицензии. **LXF**

# Лучший дистрибутив 2012

Попробуйте их все, а потом решите, согласны ли вы с результатами жестких тестов, которые провел Маянк Шарма в поисках лучшего релиза года.



## Наши участники

- » Ubuntu против Mint с. 35
- » Tumbleweed против Arch с. 36
- » Fedora против Sabayon с. 37
- » Mageia против Rosa с. 38

**Н**аше ежегодное состязание дистрибутивов уже становится традиционным здесь, в Башнях LXF. Мы делаем это, потому что любим дистрибутивы — любим их разнообразие и то, что они так меняются на протяжении года.

Но если бы мы ограничили наше сравнение одними и теми же старыми превосходящими всех колоссами, наш ежегодный парад победителей стал бы в большей степени походить на политическую осцилляцию, а не на отражение развития дистрибутивов Linux. Вот почему в этом году мы решили сделать нечто совершенно новое. Мы не хотели ограничиваться теми же самыми старыми тестами и испытаниями — мы хотели поддержать дистрибутивы, которым, возможно, не уделялось должного внимания, в то же время не забывая об очевидных преимуществах использования популярного дистрибутива.

С этой целью мы решили объединить претендентов по группам не по принципу их успеха, но по функциональности, и посмотреть, как они будут вести себя в смертельном поединке один на один. Mint схватится с Ubuntu в античной битве отца с сыном. OpenSUSE Tumbleweed схлестнется с самым популярным обновляемым дистрибутивом, Arch.

## Старожилы против новичков

Рабочие столы Gnome борются за превосходство в схватке Fedora против Sabayon, и мы протестировали лучший KDE, рассмотрев Mageia 2 и Rosa — два родственных дистрибутива, которые должны получить свою оценку. Это далеко от точной науки, потому что мнений здесь будет столько же, сколько имеется дистрибутивов. Но это будет забавно. Так пусть же начнутся игры!

# Первый раунд

## Ubuntu 12.04 против Mint 13

Ubuntu был не первым дистрибутивом, нацеленным на пользователей настольных ПК, но наиболее успешным. Аналогично, Mint не был первым дистрибутивом, попытавшимся взять все лучшее у Ubuntu и объединить это с более широкой экосистемой открытого кода, но он, бесспорно, самый успешный.

Однако за долгие годы Ubuntu расширил свой подход, нацелившись на другие области биосферы Linux – серверы и настольные системы в сфере бизнеса. Он поставляется в таком виде, что пользователи могут подстроить его под свои потребности. Например, парой щелчков можно превратить обычный дистрибутив Ubuntu в десктоп уровня предприятия, или добавить репозиторий PPA (например, Zentyal), и получить мощный сервер для бизнеса. Гибкость во имя победы!

Дистрибутив не игнорирует и новых пользователей компьютера. Его программа-установщик по-прежнему лучше: она снимает проблему деления диска на разделы, сама находит и устанавливает проприетарные дополнения в процессе установки.

После установки в дистрибутиве имеются все приложения, нужные для работы среднестатистического пользователя компьютера. Если вам нужны дополнительные программы, в Ubuntu имеется превосходный Software Centre, со всеми наворотами, вроде рейтингов и обзоров программ.

### Внешний вид

Интеграция с рабочим столом – еще одна сильная сторона Ubuntu, и свидетельством тому является MeMenu. Начинаящие пользователи также оценят унифицированные System Settings, которые облегчают задачу настройки системы. Есть здесь также и постоянно совершенствующийся облачный сервис Ubuntu One, который позволяет пользователям автоматически копировать и синхронизировать свои файлы между несколькими устройствами.



➤ Большая часть работ над 12.04 была посвящена повышению удобства Unity, а не вводу новых функций, и это хорошая смена курса для сорок из Canonical, обожающих все блестящее.

И хотя рабочий стол Unity с каждым релизом становится удобнее, все же он не стал привычным элементом для каждого пользователя. В недавних тестах мы заметили, что только новые пользователи Linux благосклонно реагируют на этот рабочий стол, и то используя его на тачскрине. Опытные пользователи рабочего стола Linux пока не привыкли к радикальным изменениям дизайна в Unity, хотя Ubuntu 12.04 сейчас неплохо работает на устройствах с несколькими мониторами.

Также приятным дополнением является новый Heads Up Display (HUD) в Ubuntu 12.04. Unity в паре с HUD значительно повысили удобство рабочего стола. Однако рабочий стол по-прежнему остается слишком ориентированным на клавиатуру. Мы ценим это, так же, как и те, кто изначально не принимал Unity. Но если наши тесты хоть в малейшей степени являются показательными, среднестатистический пользователь настольного ПК не лю-

бит жонглировать устройствами ввода. Поэтому неудивительно, что большинство не обращают внимания на хитроумный HUD.

Linux Mint является всем тем же, чем является Ubuntu, и кое-чем еще. Он берет самые лучшие инструменты из Ubuntu (например, его программу установки) и целый набор индивидуальных инструментов. Единственное, чего нет в Mint – это рабочего стола Ubuntu, Unity – и для большинства пользователей это основной довод в пользу Mint.

Итак, в то время, как Ubuntu пошел дальше с Unity, невзирая на неудовольствие пользователей, Mint обратил на это неудовольствие внимание и постарался найти решение. Эту задачу осложнил рабочий стол Gnome, отказавшись от привычной навигации по рабочему столу, но упорство и настойчивость команды разработки Linux Mint помогли ей забить победный гол в виде Mint Gnome Shell Extension (MGSE), слоя рабочего стола поверх Gnome 3.

### Мой дружище MATE

MGSE имеет нижнюю панель, меню приложений, видимые значки лотка и прочие функции, которых нет ни в Unity, ни в Gnome 3. Также в Mint 12 имеется экспериментальное ответвление традиционного рабочего стола Gnome 2, MATE. Этот рабочий стол развивается гигантскими темпами и является одним из основных преимуществ в самом последнем релизе, Linux Mint 13. Поскольку программисты Mint активно участвуют в его разработке, рабочий стол MATE отлично работает со всеми пользовательскими инструментами Mint.

Благодаря этим достижениям Mint 13 сумел взять верх над Ubuntu.



➤ Рабочий стол Mint, Cinnamon, настолько хорош, что его начинают перенимать другие дистрибутивы.

**Победитель:**



# Первый раунд

## OpenSUSE Tumbleweed против Arch

Учитывая огромное количество программ с открытым кодом, философия «релизы раньше и чаще» является трендом сезона, и этот метод находит свое логическое завершение в скользящем релизе, обновляемом постоянно, а не каждые полгода. Однако обновляемые релизы приняты не всеми. Этот поединок между одним из самых первых обновляемых дистрибутивов, Arch, и одним из самых новых, OpenSUSE Tumbleweed, стал полной неожиданностью.

Прежде всего, Arch Linux оказался самым быстрым дистрибутивом. В Сравнении менеджеров пакетов в LXF159, Arch обошел своих соперников со значительным отрывом. Arch Linux – любимый дистрибутив тех, кто предпочитает полный контроль над всеми компонентами программ на своем компьютере. Однако за этот контроль приходится расплачиваться.

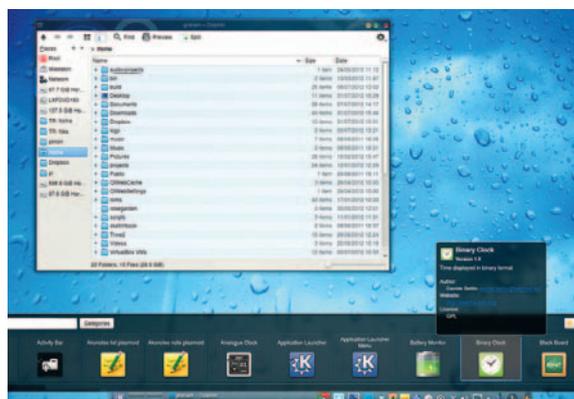
Во-первых, у Arch нет графической программы установки. Вместо этого, установка Arch требует ручного редактирования файлов настройки для получения именно того результата, который вам нужен. Единственным утешением является wiki

Arch Linux, где содержатся подробные инструкции, которые проведут вас по всем этапам процесса установки. А во-вторых, чтобы установить дистрибутив, вам понадобится интернет-соединение, причем весьма-таки широкополосное.

И вот здесь на сцену выходит OpenSUSE Tumbleweed. О Tumbleweed было объявлено в догонку релизу OpenSUSE 11.4, и хотя это отличный обновляемый дистрибутив, он остался в тени и не получил того признания, которого заслуживает.

### Мистер Tumble

Если вы о нем не слышали, Tumbleweed – это репозиторий OpenSUSE, который предоставляет скользящие обновления. Популярные дистрибутивы традиционно используют обновляемые репозитории для проверки пакетов, которые должны составить их следующий стабильный релиз.



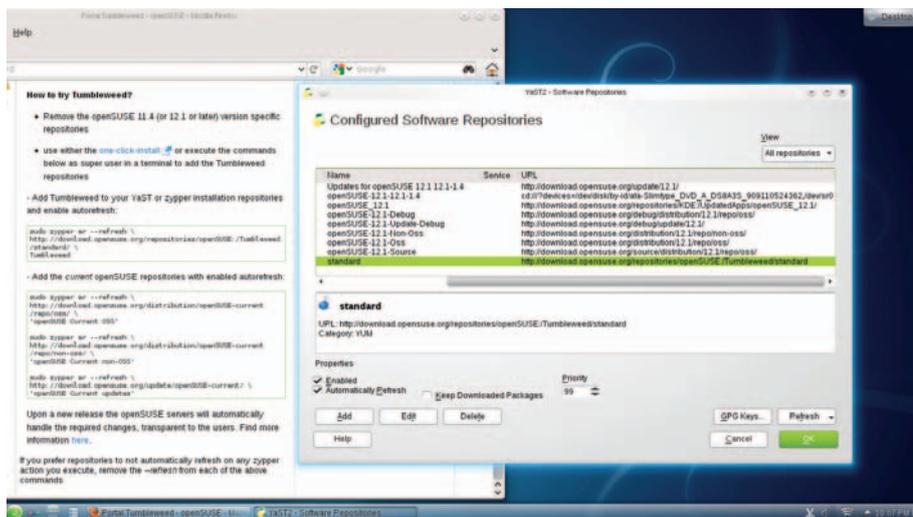
➤ Если ваш дистрибутив постоянно обновляется сам, вам не придется думать о номере версии – волшебство творится в тайне.

Примерами тому служат OpenSUSE Factory и Fedora Rawhide.

Эти репозитории имеют самые современные программы, но они не предназначены для использования в качестве постоянных дистрибутивов рабочего стола. Tumbleweed отличается от них тем, что содержит стабильные программы.

И он все хорошеет. Использовать Tumbleweed – все равно что гулять по парку. Поскольку в основном это – просто репозиторий, все, что от вас требуется – установить стабильный релиз OpenSUSE и переключиться со стабильных репозиториях на репозиторий Tumbleweed с помощью прославленного метода установки одним щелчком от OpenSUSE. Итак, в отличие от прочих обновляемых дистрибутивов, в Tumbleweed вы получаете дополнительный бонус в виде графической программы установки OpenSUSE.

Если вы не хотите месяцами ждать самого свежего релиза своих любимых программ, а ваш Интернет достаточно широкополосный и справится с постоянным потоком обновлений, то OpenSUSE Tumbleweed, победитель этой схватки – для вас.



➤ Tumbleweed предоставляет графическую привлекательность и стабильность популярного релиза OpenSUSE с остро современными программами обновляемого релиза. Полная и безоговорочная победа.



## Специальный уклон

Помимо основных дистрибутивов, упомянутых в этом материале, сообщество предлагает немало специализированных, настроенных под любые требования.

Если вы ищете дистрибутив-легковес для придания сил машинам, страдающим истощением ресурсов, попробуйте молниеносный Puppy Linux. В нем есть пользовательские инструменты для практически любых задач. Затем имеется Bodhi Linux, со своим сетевым AppCenter. И, наконец, есть основанный на LXDE Lubuntu, новый член семейства Ubuntu.

Большинство популярных дистрибутивов подойдут для десктопа уровня предприятия, но если вам нужна специальная разработка для этой цели, обратитесь

к Scientific Linux – он постепенно отвоевывает позиции у CentOS, еще одного бизнес-варианта. Если вам нужен дистрибутив с акцентом на безопасности, взгляните на Network Security Toolkit. Он собрал массу инструментов из списка [sectools.org](http://sectools.org) Top 100 Security Tools и использует и поддерживает репозиторий проверенных инструментов безопасности.

И не забудьте об интересных дистрибутивах Linux, обязанных быть в инструментари каждого пользователя Linux – CloneZilla для клонирования диска, PartedMagic, где есть инструменты управления диском и восстановления данных, и SystemRescueCD для восстановления систем Windows и Linux.



➤ Если вы жаждете анонимности в Интернете, вас выручит Tails Linux.

# Первый раунд

## Facebook Fedora 17 против Sabayon 9

**И** Fedora, и Sabayon идут с несколькими рабочими столами, но Gnome всегда был сильной стороной Fedora. Однако не так давно он начал ронять свои стандарты.

Конечно, Fedora поддерживается компанией с миллиардными доходами, но у этого дистрибутива те же проблемы, что и у Ubuntu – он пытается потворствовать наплыву пользователей, желающих видеть дистрибутив не только настольным.

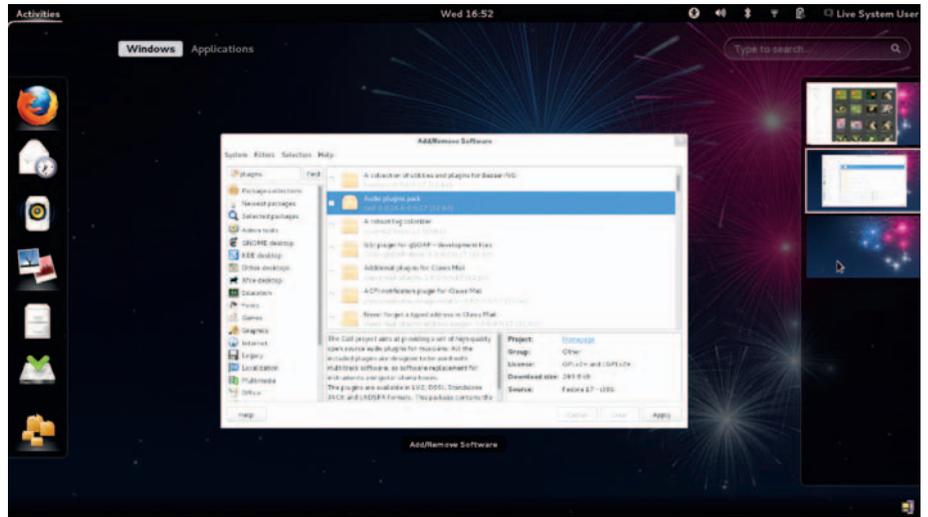
Поскольку Fedora является полигоном для испытаний Red Hat Enterprise Linux (RHEL), неудивительно, что новые релизы содержат много самых современных разработок для пользователей уровня предприятия, в том числе *virt-manager* и прочие хитрости, относящиеся к облачным технологиям.

Проблемы Fedora 17 усугубляются тем фактом, что она придерживается стандартов свободного ПО, регламентирующих компиляцию дистрибутива – а именно, без всяких включений кусков проприетарных программ, драйверов или кодеков. Это прекрасно, но не слишком годится для настольной системы. И вдобавок еще одно, что не работает на десктопе, это отсутствие достойного менеджера пакетов. *PackageKit* уже не соответствует. Если Fedora волнуют ее «настольные» пользователи, следует пересмотреть систему управления пакетами.

### Оплот настольных систем

Fedora в той или иной форме существует с 1995 года, и тот факт, что она до сих пор не утратила своих позиций в неофициальном списке популярности [Distrowatch.com](http://Distrowatch.com) является свидетельством преданности ей сообщества Linux.

Очень трудно найти у Fedora недостатки. Самым слабым местом дистрибутива была его реализация KDE, но в предыдущем релизе эта проблема была преодолена. Все же нас заставляет колебаться отрицательное отношение дистрибутива к проприетарному ПО – если бы он предоставлял



► Fedora теряет очки за пренебрежение проприетарными драйверами. В идеальном мире, конечно, нет места MP3 или Flash, но в жизни иногда нужны компромиссы.

более легкий способ его установки, как Mint или OpenSUSE, то пользователь мог бы легко сделать осмысленный выбор, вместо вынужденного. Сравните это с Sabayon 9, поставляемым со множеством проприетарных драйверов и модулей расширения для воспроизведения контента во всех форматах.

Тем не менее, Fedora – ведущий дистрибутив на базе Gnome. По этой причине с Fedora 17 вы по-

ко не все приложения в Fedora 17 придерживаются этих новых стандартов. И один из тех, кто их уже осваивает – это новое приложение Documents, которое использует инструмент индексации Tracker.

### Мир химии

Приняв во внимание критические замечания по поводу удобства своего родительского дистрибутива, Sabayon 9 заменил свой рабочий стол *Sulphur* [сера] на набор инструментов для управления пакетами *Entropy* с новым менеджером пакетов под названием *Rigo*. Он довольно-таки многословен и общается с пользователями на вполне человеческом языке.

## «Fedora следует пересмотреть свое управление пакетами.»

лучаете самый последний Gnome 3.4, а Sabayon 9 содержит более старую версию Gnome 3.2. Gnome 3.4 серьезно изменил дизайн приложений. Одна-

Невзирая на свой явный недостаток в виде устарелого релиза Gnome, здесь Sabayon 9 опять превосходит Fedora 17 внесением небольших отладок в пользовательский интерфейс Gnome. Например, окна в Sabayon 9 отображают кнопки минимизировать и развернуть, которых вы не найдете в обычном релизе Gnome, вместе с пунктом Power Off, чье отсутствие в стандартном рабочем столе Gnome 3 долгое время досаждало нашему бедному DVD-жокею Бену, из-за его сломанной руки, пострадавшей в путешествии на яхте от солнечных берегов Лигурии вослед за неряшливой супермоделью.

Но Sabayon позаимствовал у проекта Fedora один крайне важный компонент. Хотя Sabayon и основан на почтенном дистрибутиве Gentoo – точнее, на двоичной версии Gentoo – программа его установки базируется на *Anacoda*, программе установки Fedora. Хотя и с малым перевесом, но Sabayon 9 все же побеждает Fedora 17.



► Настольные пользователи оценят уведомления *Rigo*, одновременно и забавные, и информативные – в отличие от крайне лаконичных сообщений, которые порой выдает менеджер пакетов Fedora.

**Победитель:**



## Первый раунд

### Mageia 2 против ROSA

**М**ageia, которой всего два релиза от роду, относительный новичок, но создают ее ветераны — разработчики Mandriva. И у ROSA ситуация сходная. Она разрабатывается в ROSA Labs, в Москве, и это ее первый независимый релиз. ROSA Labs были партнерами Mandriva в работе над ее последним релизом, Mandriva 2011, но с тех пор их пути разошлись. ROSA 2012 — плод их дальнейших трудов.

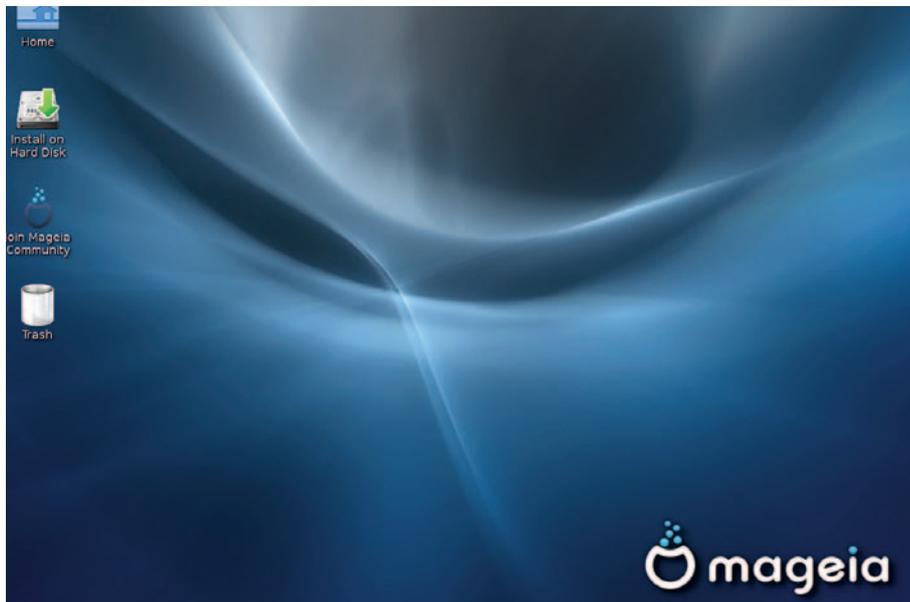
Есть две редакции ROSA 2012. Редакция Free edition не включает несвободных приложений — их можно найти в редакции Extended. За установку дистрибутива отвечает слегка приглаженная, но весьма знакомая программа установки Mandriva; для пользователей, имеющих опыт разбиения диска на разделы, она будет невероятно проста.

Помимо улучшений скрытых аспектов дистрибутива, ROSA также ввела немало количество пользовательских приложений. Одним из таковых является *ROSA Media Player (ROMP)*, который не только воспроизводит видео, но и использует программу записи действий на рабочем столе. Когда активируется функция записи, ROMP записывает видео вашего рабочего стола в формате WebM, во многом подобно встроенной функции записи, представленной в Gnome 3.4.

Затем идет *KLook*, также интегрированный с другим инструментом под названием *StackFolder*, виджетом KDE, используя который, можно просматривать содержимое папки из самой панели.

### Магия рабочего стола

С помощью *KLook* осуществляется предпросмотр файлов, аналогично функции Mac OS X, *Quick-Look*, — как из *StackFolders*, так и из менеджера файлов *Dolphin*. Здесь мы бы хотели улучшить момент и сообщить вам, что, хотя в ROSA имеются рабочие столы KDE и LXDE, ее инновации прежде всего касаются рабочего стола KDE. Одна из таких инноваций — подвергшийся редизайну рабочий



► Для достижения достойного результата команде Гаэля Дювала [Gaël Duval] из бывших разработчиков Mandriva понадобилось время, но сейчас Mageia с ходу предоставляет потрясающие возможности работы в KDE.

стол KDE, который позаимствовал ряд элементов у рабочих столов Unity и Gnome 3. В нем имеется Dash-подобный экран SimpleWelcome, который группирует приложения по их функциям, а стандартный *KDE Launcher* заменен на *ROSA RocketBar*. Дистрибутив гордится сервисом *ROSA Sync*, который находится на бета-стадии разработки, и предлагает 2 Гб бесплатного хранилища для ваших резервных копий и синхронизированных файлов.

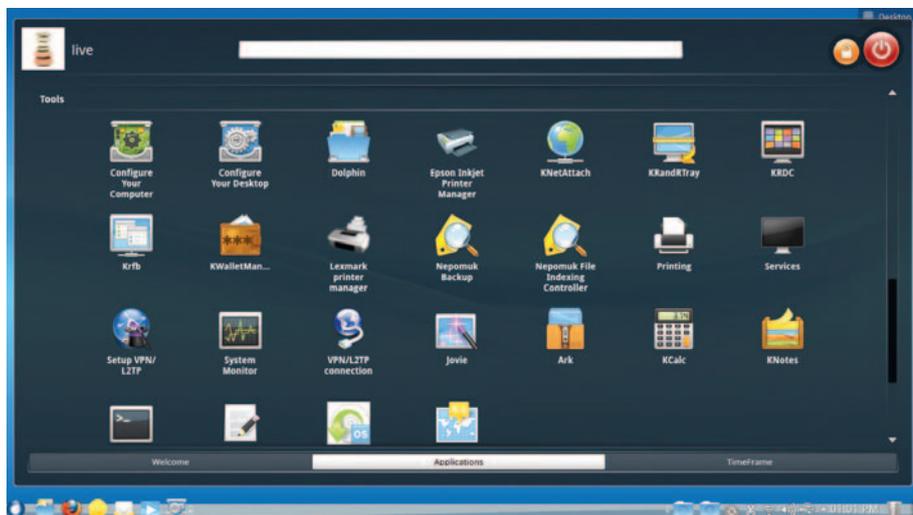
Большинство видимых функций и инструментов ROSA 2012 дебютировали в Mandriva 2011. Но поскольку Mageia отделилась от Mandriva до релиза 2011, ее самый последний релиз — Mageia 2 — во многом выглядит, как стандартный дистрибутив KDE с инструментами рабочего стола

Mandriva. Для выполнения простых задач по администрированию новичкам здесь послужит всеми любимый *Mageia Control Center*. В Mageia 2 также имеется настроенная панель с менеджером задач в виде значков, а также меню в классическом стиле.

### Не много ли инноваций?

Несмотря на акцент на рабочем столе KDE, здесь следует отметить, что Mageia 2 идет с большим количеством рабочих столов, чем ROSA 2012. Прошлый опыт с KDE 4, Gnome 3 и Unity показал нам, что пользователи Linux не любят кардинальных изменений на своих рабочих столах. Именно поэтому самая примечательная функция ROSA 2012 может отпугнуть пользователей — ROSA очень сильно модифицировала KDE, и теперь у него больше общего с Unity и Gnome 3, чем с самим KDE. Также дистрибутиву стоит отладить свой облачный сервис, который интегрирован не так хорошо, как Ubuntu One в Ubuntu.

Когда речь идет о дистрибутивах на базе Mandriva, мы должны принимать во внимание их непростое прошлое — в особенности когда претенденты делили сотрудников с исходным дистрибутивом. И хотя нам нравится то, что ROSA сделала с рабочим столом KDE, этот конкурс должен выбрать Mageia, по причине ее постоянных изменений, стабильного графика релизов, солидной инфраструктуры, и нашей слабости ко всему французскому.



► Мы пересмотрим наше мнение о ROSA, когда в следующем году появится Desktop edition. А пока что это — эквивалент мира KDE рабочему столу Ubuntu Unity.

**Победитель:**



# Полуфинал

## Linux Mint против Sabayon 9

**С**ила Linux Mint — в его здравомыслящем руководстве. Его ведущий разработчик, Клеман Лефевр [Clement Lefebvre], написал в блоге: «Мы знаем, чего хотим, и знаем, как это реализовать; и мы произведем то, что нам нужно, используя готовые компоненты, адаптируя их или создавая собственные».

Mint 13 — свидетельство исполнения этого обещания. Он включает всего понемногу. Вы получаете стабильность дистрибутива-родителя и его статус Long Term Support [Долгосрочной поддержки], лучшую программу установки, парочку свежиспеченных рабочих столов и плеяду инструментов пользователя. Mint 13 выходит в двух редакциях. Одна базируется на относительно зрелом рабочем столе MATE, ветке серии Gnome 2.x. Другая основана на подающем надежды рабочем столе Cinnamon, домашней разработке Mint, которая использует Clutter и Gnome 3 и для нормальной работы требует ускоренной графики.

Оба рабочих стола предоставляют вам знакомую рабочую среду, благодаря инструментам Mint. А для установки пакетов есть *Mint Software Manager*, который предшествует *Ubuntu Software Center*. В общем и целом, Mint 13 — истинный рай для пользователя рабочего стола.

Проект Sabayon стартовал, чтобы дать простым смертным шанс прикоснуться к совершенству Gentoo. Sabayon 9 продолжает эту традицию, и в нем многое унаследовано от Gentoo.

Дистрибутив представляет пакеты от инициативной группы Gentoo Hardened, которая поддерживает несколько проектов для Gentoo, ориентированных на безопасность. Поверх этой супер-безопасной базы предлагается на выбор три топпинга — Gnome 3.2, KDE 4.8, и Xfce 4.10.

Также в этом релизе дебютирует минималистский менеджер пакетов *Rigo*. Это выход для начинающих пользователей рабочего стола, которых отпугивают таинственные сообщения и уведомления, выдаваемые другими менеджерами пакетов. Фактически, он разруливает ситуацию



► Sabayon 9 храбро сражался, но все же проиграл грозному противнику в лице Mint 13. И в этом нет ничего постыдного.

с потенциальной слабостью этого дистрибутива, который поставляется с поврежденным пакетом *LibreOffice*. *Rigo* постоянно отправляет сообщения из репозитория Sabayon, и одно из первых содержит в себе простое решение в одну строку, позволяющее исправить *LibreOffice*.

чанию дистрибутив устанавливает *Chromium* вместо *Firefox*, что несколько странно, но опять же легко поправимо визитом к менеджеру пакетов *Rigo*.

Работа с новым менеджером пакетов — опыт кисло-сладкий. Он лучше своего предшественника, но все же имеет недостатки. Вам придется вручную перезапускать

менеджер пакетов, чтобы начать поиск в только что обновленных репозиториях.

*Rigo* основан на поиске и не выводит списков предварительно определенных категорий.

Это устроит пользователей, знающих название нужного пакета, но будет абсолютно бесполезно для тех, кто с ним не знаком.

### «Sabayon дал смертным шанс коснуться совершенства Gentoo.»

Дистрибутив кишит приложениями; среди них — браузер *Google Chromium*, медиа-плеер *VLC* и медиа-центр *XBMC*. На сайте дистрибутива вы найдете массу документации, в том числе пошаговое руководство по установке и подробный раздел FAQ, который стоит прочесть перед тем, как обращаться на форумы. Проблемы с Sabayon минимальны, и часть их легко устранить. По умол-

**Победитель:**



## Грядущие разработки дистрибутивов

Со всеми своими последними релизами, которые нам предстоит протестировать и оценить, дистрибутивы Linux в поте лица трудятся над будущими версиями. Первым пунктом в повестке дня популярных дистрибутивов стоит необходимость убедить пользователей в том, что их дистрибутивы будут работать на новом оборудовании *SecureBootEnabled*, которое появится в продаже после выхода Windows 8.

Поэтому большинство изменений в пробных релизах Ubuntu 12.10, Quantal Quetzal [Квантовый Кетсаль], остались за кулисами. На рабочем столе вы можете рассчитывать на улучшенный поиск приложений и интерфейс после его выхода в октябре.

А тем временем Linux Mint 13 выпустил свою редакцию *Xfce*, а также версию Linux Mint Debian Edition

с рабочими столами Cinnamon, MATE и *Xfce* — которая основана на Debian вместо Ubuntu.

Кстати о Debian: его следующий релиз, Wheezy, был заморожен — а значит, вместо интеграции изменений разработчики сконцентрируются на улучшении стабильности дистрибутива до его выхода в начале следующего года.

Возможно, дистрибутивом с наибольшим количеством изменений можно назвать Fedora 18. Изменения будут и неоднозначные — такие, например, как поддержка *UEFI SecureBoot* и новая политика, при которой для установки некоторых обновлений потребуются перезагрузка системы.

Однако настольные пользователи могут рассчитывать на более ровную работу благодаря функции

под названием Initial Experience. Помимо путешествия по Gnome 3, она проведет пользователей по основным настройкам системы, для которых разработчики дистрибутива не смогли обеспечить хороших предложений по умолчанию, таких, как создание учетной записи пользователя, соединение с сетью Wi-Fi, настройка учетных записей онлайн, и т.д.

Можно также ожидать значительных изменений в грядущем релизе OpenSUSE 12.2 — его выход перенесен с середины июля на сентябрь, поскольку проект завершает свой процесс разработки.

Наконец, еще одна победа Linux в целом: Mandriva вручила ключи от дистрибутива Linux-сообществу, которое выразило желание сотрудничать с сообществом Mageia.

## Полуфинал

### Mageia против OpenSUSE

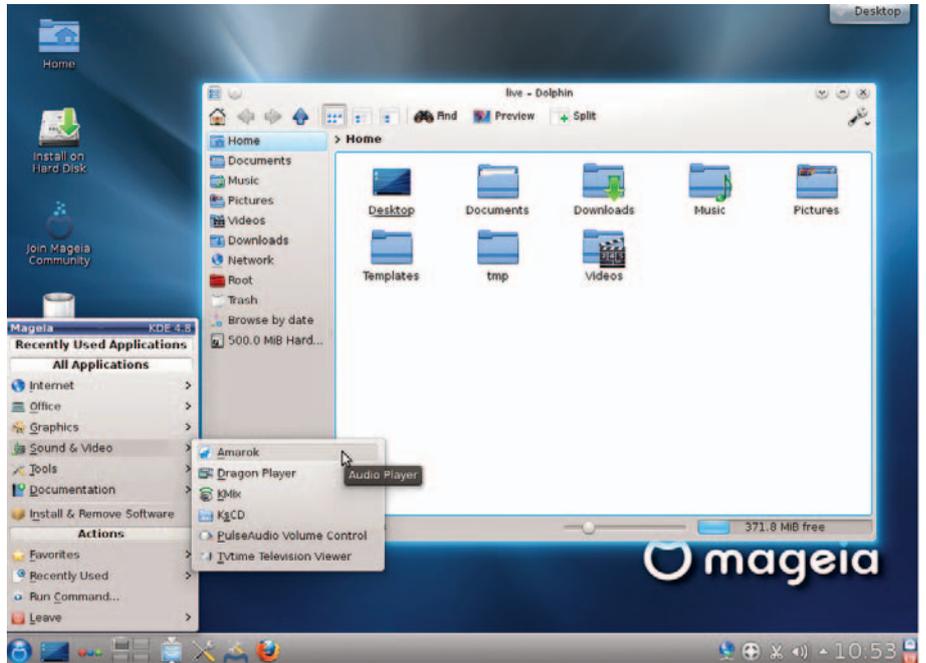
**М**ageia не снабжена проприетарными кодеками, но это легко поправить. Основной репозиторий включает только свободные программы, но репозитории с проприетарными программами добавлены по умолчанию, и активируются одним щелчком.

Все это достижимо благодаря пользовательским инструментам, начиная с набора графических инструментов для управления пакетами, с которыми легко добавлять программы и медиаисточники и обновлять систему. Лучшая функция для пользователя настольного ПК в том, что весь дистрибутив можно обновить до нового релиза из него самого, опять же одним щелчком.

Mageia 2 включает все простые в использовании инструменты конфигурации, принесшие популярность Mandriva, в том числе *Mageia Control Center* или *drakconf*, *drak3D* для настройки *Compiz* и модуль родительского контроля *drakguard*.

Mageia 2 доступна с несколькими рабочими столами, включая KDE 4.8.2, Gnome 3.4 и Xfce 4.9. Она поставляется со всеми приложениями, полагающимися настольному дистрибутиву, включая стабильные релизы с расширенной поддержкой [Extended Support Releases] *Firefox* и *Thunderbird*.

Репозиторий OpenSUSE Tumbleweed органично вписался между его основными, хоть и не новыми стабильными репозиториями. Этот репозиторий обеспечивает получение проверенных и стабильных релизов всего. До появления Tumbleweed обновляемые дистрибутивы почти исключительно предназначались для опытных пользователей Linux, поскольку единственными настоящими обновляемыми дистрибутивами со стабильными программами были Gentoo и Arch. Tumbleweed проще в использовании, благодаря графической программе установки OpenSUSE. А так как это по сути репозиторий, вы можете вернуться к стабильному релизу, заменив репозитории Tumble-



► Mageia снабжена различными рабочими столами, в том числе KDE 4.8.2, Gnome 3.4 и Xfce 4.9. Она отражает атаку OpenSUSE в нашем втором полуфинале.

weed стабильными. Плюс к тому, при переходе на обновляемый дистрибутив вы можете пометить некоторые стабильные программы, которые будут оставлены как есть, когда остальная часть дистрибутива будет обновляться.

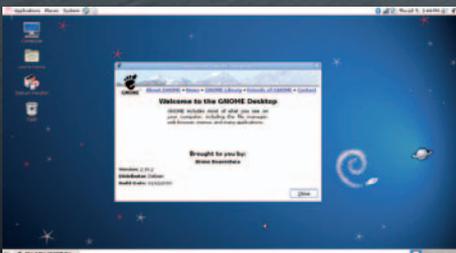
Мы слышали рассказы об обновляемых релизах дистрибутивов, которые вносили истинный хаос в систему. Увы, их не отнесешь к городским легендам. Если вы используете проприетарные драйверы, вам стоит держаться подальше от Tumbleweed, поскольку одним из компонентов, часто берущих обновления из репозитория Tumbleweed, является ядро Linux. Если вы используете

проприетарные драйверы, которых нет в основном ядре, вам придется перекомпилировать их из исходника при каждом обновлении ядра. Разработчики Tumbleweed не рекомендуют использовать обновляемые релизы в системах, требующих наличия проприетарных видео драйверов Nvidia или ATI, а также в тех, где требуется работа с *VirtualBox*.

**Победитель:**



## Полдюжины аутсайдеров отборочного тура



### Debian 6

Debian имеет двухгодичный цикл релизов, и столь долгий период созревания означает, что программы в этом дистрибутиве намного старше программ в других дистрибутивах. Однако вы можете добавлять репозитории и обновлять их, следуя инструкциям в **LXF149**, и многие считают Debian намного стабильнее популярных дистрибутивов.



### Gentoo

Один из самых популярных обновляемых дистрибутивов, Gentoo, идеален для пользователей, которые жаждут всеобъемлющего контроля за своей системой и готовы пожертвовать удобством двоичных пакетов. Обратной стороной этой медали является процесс установки, настолько хитроумный, что мы посвятили ему целое руководство в **LXF154**.



### SimplyMEPIS 11

Основан на стабильном Debian и последний раз вышел в мае 2011 как live DVD. Flash и другие проприетарные кодеки уже установлены. Программа установки – явно не для всех, но инструменты пользователя, типа *Mepis Welcome Centre* для активации репозитория сообщества на KDE, делают Mepis достойным соперником.

# Финал

## Mint 13 против Mageia 2

**В** этом турнире сошлись настоящие бойцы. В борьбе за выход в финал два наших участника победили в жестком состязании таких тяжеловесов, как Ubuntu и Fedora, и таких одухотворенных диссидентов, как ROSA, OpenSUSE Tumbleweed и Sabayon.

Итак, мы все готовы к волнующему финалу: бывалая, но не крупная рыбка в лице Mint 13 и перегруппировавшийся ветеран – Mageia 2.

Команда Mageia впряглась в работу сразу после отделения от Mandriva. Вместо того, чтобы сделать просто ответвление дистрибутива, проект уверил всех: у него есть твердые основания считать, что он имеет иммунитет к проблемам, сгубившим Mandriva.

Среди их действий было учреждение некоммерческой организации для управления совместным проектом, собственность на который принадле-

жит сообществу. Однако всего лучше то, что они не позволили бумажной работе помешать трудам над дистрибутивом. И если первый релиз в общем не сдвинулся с того места, где был на момент разрыва с Mandriva, с выходом Mageia 2 дистрибутив обрел самостоятельность.

Его команда из более чем 100 сотрудников и помощников создала гигантский дистрибутив, который устроит пользователей с самым разным уровнем навыков и опытом работы на рабочих столах всех видов. Mint 13 – это кульминация многомесячной интенсивной подготовки и экспериментальной работы разработчиков в поисках решения для пользователей, разочарованных решением Gnome и Ubuntu отказаться от классической серии Gnome 2.x.

И хотя расширения MGSE для Gnome 3 неплохо работали в качестве заплатки для Linux Mint 12,

дистрибутив активно искал лучшее решение. Рабочие столы от Mint 13, MATE и Cinnamon, отлично дополняют друг друга, и приносили дистрибутиву выигрышные очки на протяжении всего турнира. Рабочие столы, вместе с набором пользовательских инструментов, предлагаемых Mint для всех задач рабочего стола, включая основанную на WUBI программу установки Windows, делают его грозным соперником.

### Mint значит бизнес

Однако в Mint есть нечто большее, нежели стремление удовлетворить пользователей рабочего стола. Помимо релизов со множеством кодовых, этот дистрибутив также предлагает редакции без них, что делает их законными для распространения практически во всех странах мира.

Хотя это, по сути, дистрибутив одного человека, проект является жизнеспособным деловым предприятием. Он производит OEM-образы для тех, кто производит и продает компьютеры, делая для них возможной предустановку Linux Mint. При первом запуске дистрибутив проводит пользователя через простую процедуру создания учетной записи. Ждите похожей настройки в компьютерах mintBox от CompuLab. **LXF**



Кто сказал, что 13 — несчастливое число? Mint 13 — наш чемпион-2012 среди дистрибутивов, победивший Mageia 2 в эпической финальной схватке.



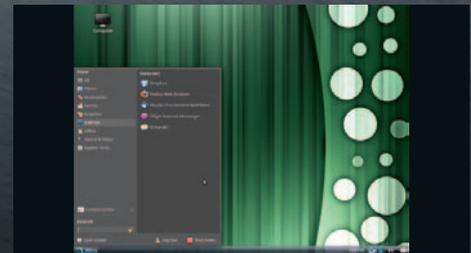
### Chakra Linux 2012.05

Chakra – обновляемый дистрибутив, популярный благодаря своей интерпретации KDE. Текущая версия идет с инструментом управления пакетами только на базе CLI, поскольку на момент выхода релиза разработчики не справились с проблемами GUI. Простая в применении система Chakra Software Bundling предлагает ряд популярных приложений.



### Zorin OS 6

Разработанный для пользователей Windows, обдумывающих переход на Linux, этот дистрибутив на базе Ubuntu доступен в нескольких версиях и идет с предустановленными Wine и PlayOnLinux. При помощи пользовательских инструментов дистрибутива можно сделать так, чтобы он напоминал Windows 7, XP, Vista, 2000, или Mac OS X.



### SolusOS 1.1

Созданный разработчиком Linux Mint Debian Edition, дистрибутив включает проприетарные коды. Текущая версия основана на Debian Stable, но следующая версия, SolusOS 2, будет основана на ветви Debian Testing, Wheezy, и включит модифицированный Gnome 3, обеспечивающий возможность работы в стиле Gnome 2.x.

# Глас народа

Эндрю Грегори беседует с Дэвидом Кобиа, техническим руководителем Ushahidi — открытой картографической платформы, дающей право голоса любому жителю Земли.



Карты — это здорово. Информация — тоже здорово. Сведите их вместе — и будет здорово в квадрате. Создайте

платформу, где каждый сможет сам сделать карту и добавить свои данные — будь то какая-нибудь ерунда, вроде путевода по местным магазинчикам, с указанием цены на пиво, или нечто жизненно важное — например, расположение источников питьевой воды — и вы дадите людям великое орудие.

Именно этим и занимается Дэвид Кобиа [David Kobia], в качестве технического лидера проекта Ushahidi ([www.ushahidi.com](http://www.ushahidi.com)), некоммерческой ИТ-компании, цель которой — вернуть информацию в руки народа, раз СМИ уже верить нельзя. Мы встретились с ним на OSCON, чтобы узнать, как продвигаются дела Ushahidi, какво это — стать Гуманистом года по версии MIT, и почему свободное ПО так важно для этого проекта.

**LXF:** Расскажите нам про Ushahidi. Правда ли, что создание проекта подтолкнули акции протеста в Кении несколько лет назад?

**Дэвид Кобиа:** Я не знаю, что именно вы об этом слышали, но в конце 2007 там состоялись выборы [подозрения в подтасовке были с обеих сторон, и конфликт перерос в насильственные действия — погибло 1500 человек и 250 000 лишились жилья]. И, естественно, СМИ тут же прикрыли: не стало ни телевидения, ни радио, ни газет. Но власти не учли, что есть Интернет и есть мобильная связь. И свидетельства о применении насилия появились в многочисленных блогах и внутри, и вне Кении. У людей появилась острая необходимость где-то хранить эти материа-

лы, а также сообщать о том, что они видят. Вот мы и создали первую версию Ushahidi [в переводе с суахили — «свидетель»]. Она была сделана за одни выходные, и в итоге у нас появился номер телефона, на который можно было прислать SMS с информацией. Я и не предполагал, что эта скромная попытка возымеет такой эффект.

**LXF:** В том числе и то, что MIT назвал вас Гуманистом года в 2010-м?

**ДК:** Это до сих пор сносит мне мозг. Сроду не считал себя гуманистом. Прямо бред какой-то.

**LXF:** Но для биографии — просто шикарно. А как все это стало технически выполнимо?

**Вы используете Google Maps и их API?**

**ДК:** Ну, главная заслуга Google Maps в том, что они двинули идею карт в народ. Они были не первыми, кто обещал карты данными, но стали самыми известными.

## О ПЛАТЕ ЗА GOOGLE MAPS

«Позиция Google понятна: они смотрят на салдо, это же коммерция.»



**LXF:** Есть же еще OpenStreetMap...

**ДК:** Верно, есть OpenStreetMap, и туда-то мы и перешли, поскольку у Google есть свои ограничения. С определенного уровня доступ оказывается платным, а открытое сообщество такой подход бесит. Наш девиз – «иметь полный доступ к ресурсу и не зависеть от чего-то вне нашего контроля». Так что теперь, выкатив собственные тэги через OpenStreetMap, мы пользуемся им в полном объеме, не оглядываясь ни на какие корпорации. Google, конечно, хорошие ребята, но всегда есть такие вещи, на которые люди реагируют болезненно. Мне позиция Google понятна: они смотрят на сальдо, это же коммерция, и есть акционеры, перед которыми они несут ответственность.

**LXF:** Применяли ли пользователи проект к другим видам информации?

**ДК:** Вообще-то главное его применение – на случай происшествий и критических ситуаций, то есть там, откуда мы начали. Ведь проще

всего сплотить народ перед лицом бедствия, когда все жаждут помочь. Если речь идет, скажем, о выборах, активистов будет меньше – эмоциональная вовлеченность уже не та.

Есть так называемое правило 1-9-90: 1% людей создает контент, 9% – что-то иногда добавляют, а все остальные просто смотрят. В кризисных ситуациях это соотношение меняется. Откликаются многие люди: «Я бы хотел(а) помочь – скажите, в чем».

Ну и, кроме кризисов, есть еще наблюдение за ходом выборов. Кто-то использует проект в образовательных целях, а кто-то и для более занимательных вещей: составляя карты преступности или выбоин на лондонских улицах, или наблюдая за массовыми сокращениями в Лидсе.

**LXF:** Как в мэрии Лидса в Англии?

**ДК:** Да, по-моему, это сделала газета *The Guardian*. Получилось довольно интересно. Вроде бы у них была даже карта забастовок в метро. А здесь, в США, нечто подобное соз-

дала во время сильнейшей снежной бури вроде бы *Chicago Tribune*.

**LXF:** Но в этом и прелесть открытых платформ. Вы создаете проект для своих целей где-нибудь в Кении, а его берут и применяют совершенно в другом месте и совсем иначе.

**ДК:** Это и есть открытость, это и замечательно. И означает, что мы на правильном пути.

**LXF:** Легко ли создать собственный проект, применив Ushahidi? Он устроен по принципу групп или сообществ, как Google Plus?

**ДК:** Мы обычно говорим, что Ushahidi – это решение лишь на 10%. Некоторые полагают, что стоит сделать сайт – и у него тут же возникнет аудитория, а на самом деле здесь есть над чем поработать, поскольку нужно обеспечить охват: требуется обучение обращению с картами и проверка достоверности поступающего материала... конечно, это задача для целой группы. Не то что один тип сидит перед мощным табло и ждет, пока появится информация.

Что касается технической базы, то здесь мы всегда равнялись на *WordPress*. Это один из популярнейших открытых продуктов в данной сфере. И по моим ощущениям, мы сейчас где-то на одном уровне, все идет неплохо, но многим пока сложно освоиться. Поэтому мы решили запустить проект *Crowdmap*. Это своего рода облачный *Ushahidi*. Вы туда заходите, создаете свой субдомен, скажем, [andrew.crowdmap.com](http://andrew.crowdmap.com), и буквально через пару минут у вас будет собственная рабочая версия *Ushahidi*. Это удобно, чтобы привлечь достаточное число пользователей. Потом, если захотите, можете перейти на свой сервер, он же бесплатный для скачивания.

**LXF:** На *OSCON* в этом году было много такого: все так или иначе рвутся в облака.

**ДК:** Знаете, слово «облачный» уже сделалось штампом.

**LXF:** Да оно вообще ничего не значит!

**ДК:** Зато удобно тем, кто не хочет ничего объяснять толком. Мне лично это слово не нравится, но иногда просто удобно бывает вздеть руки к небу и заговорить про «облако».

**LXF:** А вы видели проект, в Нью-Йорке, кажется, где составили карту цен на *Budweiser*?

**ДК:** Нет, мне он не попадался. А что, цены на *Budweiser* так сильно отличаются, что есть такая необходимость?

**LXF:** Да нет, всего на пару центов. По-моему, это был проект чисто по соседству.

**ДК:** Любит народ свое пиво.

**LXF:** Ну и им просто нравится с таким возиться. Использовался ли *Ushahidi* во время землетрясения на Гаити?

**ДК:** Вероятно, на Гаити *Ushahidi* пригодился бы никогда. Между событиями на Гаити и выборами в Кении шла большая работа над программным обеспечением, появилось много идей насчет его функционирования. Гаити во многом подтвердил для нас правильность нашего пути, в том, что возможно организовать совместные усилия для такого масштабного проекта. Но к такому не подготовишься. Тогда было задействовано множество групп, включая и *Ushahidi*.

Мы работали в связке с местной телефонной компанией – был создан специальный номер, на который любой мог отправить имеющуюся у него информацию, и через нашу программу прошло порядка 45 000 текстовых сообщений. Таких как «нужна вода» или просто «застряли под завалами».

Причем все это мы получали на креольском, и понадобилось срочно привлечь местных жителей к переводу.

Гаитян попросили создать какую-нибудь систему быстрого просмотра и перевода этих сообщений, чтобы по карте могли сориентироваться

и сказать: «Ага, это на углу такой-то улицы в Порт-о-Пренсе». А затем передать эту информацию службам спасения, чтобы они как можно скорее туда добрались.

С другой стороны, удивительно было само то, что мы получали всю эту информацию, нахо-

## ОБ ОБЛАЧНОМ ЖАРГОНЕ

**«Мне это слово не нравится, но иногда удобно бывает заговорить про “облако”.»**

дясь в береговой охране США, наносили ее на карты, чтобы где-то вдали ее использовали на деле. И для всех это был колоссальный труд.

Но главное, что можно привлечь к помощи людей, удаленных от места событий. Вот что значит виртуально находиться «в первом ряду», наблюдая за событиями мировой арены.

**LXF:** И много потребовалось времени, чтоб перевести все это с креольского на английский?

**ДК:** Сначала, когда они только начали поступать, мы ужаснулись: «Черт! Неужели в мире не все еще говорят на английском?» Но потом отыскались такие ребята – *Crowdfunder* – не знаю, как точно их описать, но тоже проект сообщества. И вместе с ними мы создали простейшую платформу – самую элементарную, только для перевода с одного языка на другой и обратно. Не знаю, сколько гаитян тогда подключилось, чтобы помочь. В какой-то момент нам пришлось скрыть от переводчиков номера телефонов, с которых поступали сообщения: они принимали это чересчур близко к сердцу. Они хотели тут же перезвонить тому человеку, чтобы сказать, что помощь уже идет, а этого от них уже не требовалось.

**LXF:** А без *Open Source* вы бы справились, как вы думаете?

**ДК:** Думаю, что нет, поскольку, во-первых, у каждого свой подход к любой технологии. Допустим, при решении вопросов безопасности – они всегда актуальны – с хранением или передачей информации, как все это делается.

Но само знание лежит выше и помимо технологической стороны. Необходимо делиться собственным опытом, чтобы он послужил и другим. Так что открытым должно быть не только ПО, но и знание в целом. Поскольку

тогда мы сможем передать то, что накоплено нами за эти 3-4 года, чтобы люди могли этим пользоваться... мы сразу приняли решение, что вовсе не обязательно всем проходить тот же путь с самого начала, сталкиваясь с проблемами, уже нами решенными. И это не только технологический компонент; знание намного важнее технологий.

**LXF:** Верно. Существуют и закрытые блог-платформы, и карты, но все зависит от духа проекта.

**ДК:** К слову, о картах: когда случилось землетрясение, ни на *OpenStreetMap*, ни на *Google* не было ничего о Гаити. Разве что основные дороги, и никаких ориентиров, и был случай, когда людей направили на мост, которого вообще не существовало. Быстро и оперативно предоставить информацию смогло только сообщество *OpenStreetMaps*.

Такой же пример был в секторе Газа, когда карты *Google* устарели, а на *OpenStreetMaps* ребята объехали окрестности и все обновили, показав, допустим, что в данном месте стоит мечеть, а не церковь. Вот что означает мобильность *Open Source*.





**LXF:** А вы всегда были сторонником свободного ПО?

**ДК:** До Ushahidi я работал в обычной компании и никогда не писал код для открытых сообществ. И меня никто не надоумил об этой идее... Конечно, я использовал Open Source – все это когда-нибудь делают, – *Drupal*, *WordPress*, *Joomla* и т.д. – некоторые проекты уже вполне зрелые. Ты с ними работаешь, но не задумываешься, сколько труда в них вложено. Так что я был просто одним из тех, кто грызет гранит нудных программ. Когда я приступил к работе над Ushahidi, который стал открытым, идея о предоставлении бесплатного ПО у меня в голове не укладывалась. Если ты пришел с той стороны, где слова «программа» и «бесплатно» и рядом не стоят, приходится во многом перечувствовать. А теперь я уже и представить не могу, что бывает по-другому.

«Открытое» мышление – это... даже слов тут не могу подобрать. Это образ жизни.

**LXF:** Сколько людей сейчас в проекте?

**ДК:** Нас 18, и живем мы примерно в девяти часовых поясах. Начиная от Сан-Франциско и штата Вашингтон на западе до Окленда в Новой Зеландии, и большая толпа в Кении. Один парень – из Южной Кореи, ну и все

**ОБ ОТКРЫТОСТИ ПО**  
**«Открытое» мышление –**  
**слов тут не подобрать.**  
**Это образ жизни.»**

остальные где-то между ними. Так что планерки по понедельникам проводить непросто: у меня 10 утра, в Новой Зеландии уже 2, а где-то еще – 6 вечера.

**LXF:** И как, получается?

**ДК:** В общем, да. Так что по понедельникам у нас общее собрание по Ushahidi, а затем по вторникам еще на часок встречаемся только с сотрудниками техподдержки. Получается неплохо, спасибо *Skype*.

**LXF:** Как вы думаете, в каких сферах Ushahidi будет наиболее востребован? Где он сможет изменить мир?

**ДК:** Его используют в небольших городах... мы стараемся развивать это направление. Множество мелких городов его используют. Мне кажется, именно они – наша целевая аудитория. Просто чтобы людям было проще «достучаться» до местной мэрии, чтобы сказать: «Эй, здесь светофор не работает», или про выбоины на дорогах, или про нужды здравоохранения. Думаю, это было бы здорово. **LXF**



# Великие споры Linux

Почему в мире Linux пылают столь страстные споры? Откуда они начались и что они означают? Майк Сондерс натягивает асбестовые штаны...

**К**акого бы мнения вы ни придерживались по любому вопросу, в Интернет вы всегда найдете желающего с вами поспорить.

Бескрайние просторы Интернета, с его чат-румами, серверами NNTP и списками рассылки, служат полигоном для бурлящих споров с сотнями спорщиков, а когда в эту смесь приносишь технологию, все это становится угрожающим. Вовек не угаснет пламя старых раздоров: зайдите куда следует, и вы найдете там тех, кто до сих пор выясняет, что же все-таки

было «подлинно» классическим 8-битным компьютером — ZX Spectrum или C64.

В мире Linux тоже кипят нешуточные страсти, однако они все же несколько более обоснованны и обдуманно (ну, как правило). Большинство из них уходит корнями в исключительно политизированную природу движения свободного ПО — здесь дело в куда большем, чем просто биты и байты. Многие полагают, что, поддерживая и отстаивая определенную операционную систему или программу, они помогают сделать мир лучше. И,

возможно, так оно и есть, но в результате все оказываются особо взвинчены.

Однако нам известно, что множество пользователей — любителей Linux не обладают достаточным досугом, чтобы вникать в детали в этих дебатах, и считают некоторые из них довольно странными. Поэтому мы решили взять самые животрепещущие темы и добраться до сути спора. Мы хотим осветить обе стороны, чтобы вы смогли вынести собственное суждение, основываясь на голых фактах, без всякого драматизма. Что ж, начнем...

# KDE vs Gnome

Два рабочих стола с диаметрально разным происхождением.

Споры вокруг темы KDE против Gnome ныне концентрируются исключительно на технических вопросах: например, какая среда рабочего стола проще в использовании, или в какой больше функций. Но раньше, в начале 1990-х, ситуация была куда сложнее. Когда объявили о выходе KDE в 1996 году, мир Linux уже не первый год ждал появления полноценного интегрированного рабочего стола. Конечно, были у нас разные приятные менеджеры окон и программы, но в плане интеграции все это было несколько неуклюже. KDE вознамерился это исправить – и осыпал многих.

Но для многих потенциальных пользователей появилась огромная проблема, а именно – выбор графического набора инструментов. Команда KDE решила использовать Qt, предоставлявший широкий выбор функций, но также и отягощенный всяческими лицензионными осложнениями (чтобы писать проприетарные программы, разработчики должны были приобрести коммерческую лицензию). С самого начала KDE стал восприниматься как противоречивый проект, не в полной мере отвечающий духу свободного ПО.

Следом, уже через год, появился проект Gnome – и он был создан на GTK, полностью сво-

бодном инструментарии. Но в то время как команда KDE писала все с нуля, разработчики Gnome в некотором смысле спешили выпустить конкурентоспособный релиз 1.0, и заново использовали уже существующие программы, такие, как менеджер окон Enlightenment. Однако KDE 1.0 и по виду, и по ощущению был намного отягощеннее и завершеннее.

Итак, с самого начала имелись политические разногласия, а затем – разница в техническом качестве. Позднее, когда Qt вышла под лицензией LGPL, первая проблема была снята, и с той поры спор сосредоточился на функциях. В общем и целом, KDE считают более сложным и требовательным, предоставляющим колоссальный выбор функций и возможностей настройки, но несколько пугающим для новых пользователей. Gnome же, наоборот, ограничивает уровень контроля со стороны пользователя, что благоприятствует наработке опыта постоянной работы с машиной, но отчасти разочаровывает тех пользователей, которые любят подвигать все индивидуальной настройке.



➤ Gnome 1.0 выглядел сыровато, но был «свободной» альтернативой KDE.

Недавние крупные реформы сред рабочего стола подлили масла в огонь. Пользователи Gnome глумились над попытками KDE 4 преобразовать рабочий стол, похваливая стабильность серии Gnome 2.x. А спустя пару лет сами попали в то же положение, когда Gnome 3 вышел в совершенно непривычном виде. Многие пользователи в те тяжелые времена сменили позиции, и теперь питают сердечную привязанность к бывшим врагам...

# Linux vs FreeBSD

Какая система является лучшей открытой Unix-подобной ОС?

Вы, вероятно, слышали о FreeBSD, особенно если обращаетесь в мире Linux уже несколько лет. У нее много общего с Linux: это Unix-подобная операционная система с открытым кодом, разрабатываемая сообществом и работающая с KDE, Firefox, LibreOffice, Apache, MySQL и практически с любыми другими основными программами Linux, которые только на ум взбредут. Во многих ситуациях она служит прямой заменой Linux, и большинство навыков, полученных в Linux, применимы и ко FreeBSD.

Тем не менее, не утихают споры о том, которая же ОС лучше. Чтобы разобраться в этом, нам придется немного углубиться в историю. Корни FreeBSD уходят в BSD Unix, ОС, разработанную в конце 1970-х в университетских кругах. Существовало немало ответвлений BSD, и самым многообещающим из них оказалась появившаяся в 1993 году FreeBSD для ПК x86. Он обеспечивала солидную, хорошо протестированную полноценную основу Unix для потребительских ком-

пьютеров, тогда как Linux считался еще незрелым по причине своей новизны. К несчастью, ранней проблемой FreeBSD стал пресловутый камень преткновения, когда держатели авторских прав кода Unix крупно поссорились со старой командой BSD. Это все было очень запутанно и сложно, и отпугнуло немало потенциальных разработчиков от работы над FreeBSD – они переключили свое внимание на Linux. Команде FreeBSD пришлось переписать большую часть кода, однако на это ушло несколько лет, и они упустили бесценное время.

Сегодня основными источниками разногласий между сторонниками Linux и FreeBSD являются методы разработки и лицензия. Что касается первого, здесь команда FreeBSD намного сплоченнее и консервативнее, и принимает новые функции только после их тщательного тестирования. Исходный код для всей ОС поддерживается единым деревом – в противоположность Linux, чей подход к новым функциям и поддержке оборудования – сущий «дикий Запад», как и то, что ОС разраба-

тывается отдельными группами (одна для ядра, одна для библиотеки C, и т.д.).

Лицензия BSD, по сути, говорит следующее: «Делайте с кодом все, что вам заблагорассудится, но не заявляйте, что это вы его написали (то есть сохраняйте уважение к оригинальному авторским правам). Это, безусловно, намного более свободный подход, чем практикуемый в GPL, которая применяется в Linux, но это не укрепляет свободу – из кода FreeBSD возможно создать проприетарный продукт. Так что те, которые считают свободное ПО политическим движением, склоняются к использованию Linux и GPL, а те, кто рассматривает все это с сугубо практической точки зрения, разворачиваются в сторону лицензии BSD и FreeBSD.»



# Linux vs GNU/Linux

Что в имени тебе моем? – да похоже, вся история нашего фаворита.

С периодичностью в несколько месяцев мы в **LXF** получаем сердитые письма с жалобами на неупотребление в нашем журнале полного названия Linux – то бишь GNU/Linux. Та же тема периодически всплывает на форумах **LXF** или в *TuxRadar*. И знаете что? В этом есть свой резон. С точки зрения журнала, мы бы сказали, что он недостаточен силен как причина менять наш способ писать про наш основной предмет, но за этими возражениями стоит нечто важное – и большее, чем простая педантичность.

Это – вопрос исторической перспективы, и нам следует вернуться в 1983 год. В это время некто Ричард Столлмен [Richard Stallman] основал проект GNU с высокой целью создания совершенно свободной (в смысле свободы делиться кодом и модифицировать его) Unix-подобной операционной системы. То, что за этим последовало, вызывает самые горячие споры, и вот аргументы обеих сторон:

**Сторонник GNU:** «Начиная с 1983 года, разработчики GNU трудились над созданием полной операционной системы, являющейся заменой практически всем компонентам типичной системы Unix. К 1991 году они почти закончили свою работу. Им только не хватало



➤ Ричард Столлмен поддерживает термин GNU/Linux, но не контрпродуктивна ли его работа?

ядра. Финский студент по имени Линус Торвалдс [Linus Torvalds] написал ядро, которое подошло к наработкам GNU, и, приложив еще немного усилий, мы завершили операционную систему – GNU/Linux».

**Сторонник Linux:** «Начиная с 1983 года, разработчики GNU трудились над созданием замен разных частей системы Unix, но брали немалую часть готового кода из других проектов. Они проделали неплохую работу, но никогда толком не концентрировались на самом важном – на ядре. И так, к 1991 году у них хватало полезного кода в разных проектах, но все это стало нормальной ОС не раньше, чем Линус выпустил свое ядро».

## «Система Linux – не просто комбинация фрагментов GNU и Linux.»

Истина, как водится, лежит посередине. У Столлмена был конкретный план, и он не покладая рук трудился, чтобы проект GNU наконец заработал. Ранние версии того, что мы называем Linux (как цельная ОС), не появились бы без проекта GNU. Но нельзя и сказать, что в 1991 году GNU был на 99 % готов и для его завершения недоставало крошечного ядра. Ядро – наиболее важный компонент ОС и берет на себя массу работы, чтобы все шло правильно.

### Распространение слова

Люди стали обозначать термином Linux всю операционную систему, и, понятно, Столлмен был несколько обижен, что его проекту GNU перепало мало чести – вот он и предложил название “GNU / Linux”. Но это было не просто привлечение внимания.

И если Линус Торвалдс – личность исключительно прагматичная, и его интересует лишь

возможность создания полезного кода, который мог бы улучшать каждый, Столлмен рассматривал свободное ПО как политическое движение, способ сделать наш мир лучше. Поэтому наличие приставки GNU ему было нужно не для улаживания своего эго. Нет, он хотел довести до людей, что на самом деле означает GNU, почему свобода ПО так важна и почему мы должны обращать внимание не только на прагматичные проблемы.

Все это хорошо и мило, но приводит только к возникновению дальнейших споров. Например, современная система Linux – это не просто комбинация фрагментов GNU и Linux. Она включает код из X Window System, Perl и мас-

сы других проектов, не относящихся к GNU. Принимая все это во внимание, не следует ли нам именовать систему GNU/X11/KDE/Perl/XBill/Linux? Длинновато... А как

решить, какие части должны войти в «официальное» полное название? Для некоторых громадной важностью обладает *LibreOffice*, построенный на обширной кодовой базе; не звучать ли ему, как GNU/LibreOffice/Linux? Но постойте, ведь *LibreOffice* отпочковался от *OpenOffice*, который был *StarOffice*, так что... ну, вы поняли, куда нас это может завести.

Даже если оставить просто GNU/Linux, все равно будет не слишком красиво. Надо отдать справедливость Столлмену, он попытался найти альтернативу в виде “Lignux”, но и это звучит не здорово. Добавьте сюда бесчисленные варианты произношения этого названия, и вы получите полный винегрет.

И все же команда GNU продолжает бороться за это название, хотя они явно чересчур носятся с этой проблемой. У них на сайте есть FAQ по поводу наименования GNU/Linux по адресу [www.gnu.org/gnu/gnulinux-faq.html](http://www.gnu.org/gnu/gnulinux-faq.html), но, как связывал наш Эндрю Грегори в подкасте *TuxRadar*, он длиннее, чем конституция США.

GNU несет в себе важную идею свободы и возможности делиться, но многие полагают, что она могла бы распространяться куда более эффективными способами. Например, сторонники «полного» наименования могли бы не хныкать, что кто-то говорит “Linux” вместо “GNU/Linux”, а попросить создателей дистрибутивов разместить на своих сайтах яркую кнопку «Вы пользуетесь этим благодаря GNU [Brought to you by GNU]» со ссылкой на объяснение роли GNU в проекте.

## Закон Годвина

Главный катализатор споров онлайн – анонимность и обилие досуга у спорящих. Если люди могут скрыть свои личности и посвятить восемь, а то и больше часов пламенным спорам, даже самая невинная тема – ну, скажем, «Мои котятки» – способна превратиться в жуткий шабаш ненависти.

Причем существует весьма интересный феномен: чем ожесточеннее спор и чем дальше он ухо-

дит от первоначальной темы, тем сильнее вероятность, что кто-то кого-то обзовет фашистом (а то и Гитлером). Первым это заметил юрист и известный всему Интернету отличный парень Майк Годвин [Mike Godwin], отсюда и название – «закон». Участники споров частенько заранее упоминают Годвина, дабы избежать возникновения ассоциаций с Третьим рейхом.

# RPM vs .deb

Две системы управления пакетами по цене одной.

**В**се новые и новые изобретения колеса стали постоянной проблемой в мире ПО. За чем брать чей-то код, если можно написать собственный – вдесятеро лучше? Частенько разработчики делают что-то свое чисто из гордости, или потому желая узнать, как это работает. Это хорошо, и никто не запрещал пробовать что-то новое, но в итоге это означает огромное количество продублированных результатов. Именно

так и случилось с системой управления пакетами. Пакеты от Debian **.deb** появились раньше, чем RPM (от Red Hat), и некоторое время между ними существовала весьма полезная конкуренция.

В каждой системе были свои преимущества, но постепенно они обе сравнялись. Например, сторонники Debian прохаживались насчет пользователей Red Hat – мол, в последнем дистрибутиве нету чудес **apt-get**. Но тут появились **apt-rpm**,

и улыбки с их лиц исчезли. Сегодня проблема в том, что обе системы уже настолько зрелые и укоренившиеся, что ни одна не считает нужным принять чужую сторону. Администраторам Linux жилось бы намного проще, если бы мы все выбрали одну систему; но пока обе достойно справляются с работой, вряд ли мы увидим, как одна из них испустит дух.



# Emacs vs Vim

Не просто текстовые редакторы, а образ жизни.

**Е**сли вы полагали, что худшее – это войны рабочих столов, проведите-ка пару часов в пылу дебатов на тему *Emacs* против *Vim*. У этих редакторов самые упертые фанаты, которых только можно вообразить. Текстовые редакторы вряд ли можно считать самой волнующей темой для страстных споров, но авторы и программисты практически живут в них, проводя за этим долгие часы ежедневно.

Пользователи *Emacs* считают сторонников *Vim* чокнутыми, которые заучивают уйму дурацких сочетаний клавиш и не могут оторваться от редактора, созданного для модемных соединений

свыше 300 бод. Пользователи *Vim* считают сторонников *Emacs* чокнутыми, неспособными справиться с минималистским редактором, которым необходимо вводить смехотворно длинные команды вроде **M-xpaste-paragraph** для самых тривиальных операций.

Здесь также есть нечто от битвы Linux против FreeBSD. *Vi* (на нем основан *Vim*) разработал Билл Джой [Bill Joy], ключевая фигура в истории BSD, а *Emacs* написал Столлмен. Сторонники *Vi* обычно шутили, что *Emacs* – сокращенное “Eight Megabytes And Constantly Swapping” – “Восемь мегов и сплошная подкачка».



Интернет пухнет от форумов, блогов и видео, пытающихся утвердить истинного короля редакторов.

# И прочие

Текстовые редакторы, языки программирования...

**Л**юди очень привязываются к инструментам, которыми пользуются. Иногда немного раздражает, когда видишь, как кучка обитателей форума спорит о том, где же лучше писать диплом – в *AbiWord* или в *LibreOffice* (особенно когда знаешь, что идеальным решением является *LaTeX*), но зато сколько эндорфинов выделяется, если удастся кого-то убедить!

Языки программирования тоже приводят к весьма забавным перепалкам – участники вложили в них много сил и времени. Например, представьте себе, что вы потратили десять лет на освоение Python, и вдруг появляется какой-то сопляк и заявляет, что Ruby намного круче. И даже если технически это правда (мы здесь не принима-

ем ничью сторону), как все же ужасно думать, что вы потратили свое время не на тот язык, верно?

А еще есть боязнь перемен («Мне никогда не понадобится ничего, кроме Slackware!»), замеры молодечества в плане электроники («На моем Arch Firefox запускается на 0,3 мс быстрее, чем на твоём Gentoo, так что все барышни – мои»), ну, и так далее. Сколько элементов человеческой психологии и поведения отражается в этих спорах!

Ну, а что ждет нас в будущем? Уже наблюдаются занимательные баталии между *GCC* и *LLVM*, особенно по мере принятия *LLVM* некоторыми дистрибутивами как компилятора по умолчанию. Споры по поводу рабочих столов теперь идут в трех направлениях – новых сторонников набира-

ет *Xfce*, а если Wayland когда-нибудь поднимется до замены X Window System, то и тут мы ожидаем небезынтересные битвы. Будет весело! **LXF**



## Обратная связь

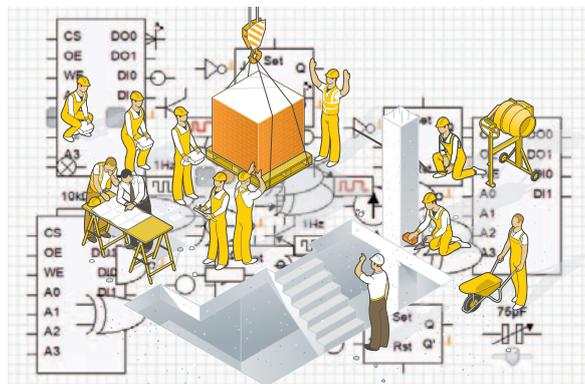
А вы когда-нибудь участвовали в подобном споре? Вам удалось убедить кого-то принять вашу точку зрения с помощью взвешенных аргументов? Или вы на каждом шагу применяете Закон Годвина?

Поделитесь с нами своими идеями по этой бесконечной теме, а также и о том, что, по вашему мнению, вызовет самые оживленные споры в будущем, по адресу: [ixf.letters@futurenet.com](mailto:ixf.letters@futurenet.com).

# CPU своими руками

## часть вторая

Еще два компонента электронного сердца вашего CPU: **Джонатан Робертс** подводит вас на шаг ближе к рабочему процессору.



**В** LXF158 мы показали вам, как построить собственный компьютер из базовых электронных блоков. По сути, мы изготовили двоичный сумматор. Мы закончили статью, сказав, что созданное нами — это еще далеко не настоящий компьютер, и вам нужны еще несколько компонентов, прежде чем ваша машина сможет претендовать на звание компьютера.

В этой статье мы рассмотрим дальнейшие шаги и представим вам еще несколько компонентов, продемонстрировав, как расширить возможности вашего сумматора до выполнения других действий (**AND [И]**, **XOR [ИСКЛЮЧИЛИ]** и **NOT [ИЛИ]**) и создать цепи памяти для хранения ввода и вывода.

Перед чтением этой статьи вам стоит прочесть предыдущую часть, чтобы освежить память, поскольку мы исходим из того, что вы с ней ознакомились. Если у вас нет экземпляра LXF158, вы сможете найти PDF этой статьи на диске этого месяца.

### Декодер

Первый компонент, который мы рассмотрим — это декодер, или, точнее, линейный декодер. Он очень прост, но очень удобен в работе и сам по себе, и декодеры также используются в мультиплексорах — что исключительно важно для остальной части нашей статьи.

По сути, все декодеры позволяют сделать выбор из множества выходов, используя только половину входов. Более конкретно — при единственном входе декодер позволит вам выбирать между двумя выходами, два входа обеспечат вам выбор из четырех выходов, и т.д. В случае декодеров мы говорим не о входе и выходе, а об адресной линии и линии передачи данных. Таблица истинности для простого линейного декодера 1-в-2 размещена ниже, а схема соединений приведена на рис. 1.

Как видите, все очень просто. Адресная линия раздвоена, и вход **NOT** размещен на одном ответвлении этого раздвоения. Таким образом, только одна из этих линий передачи данных может за один раз передать **1**.

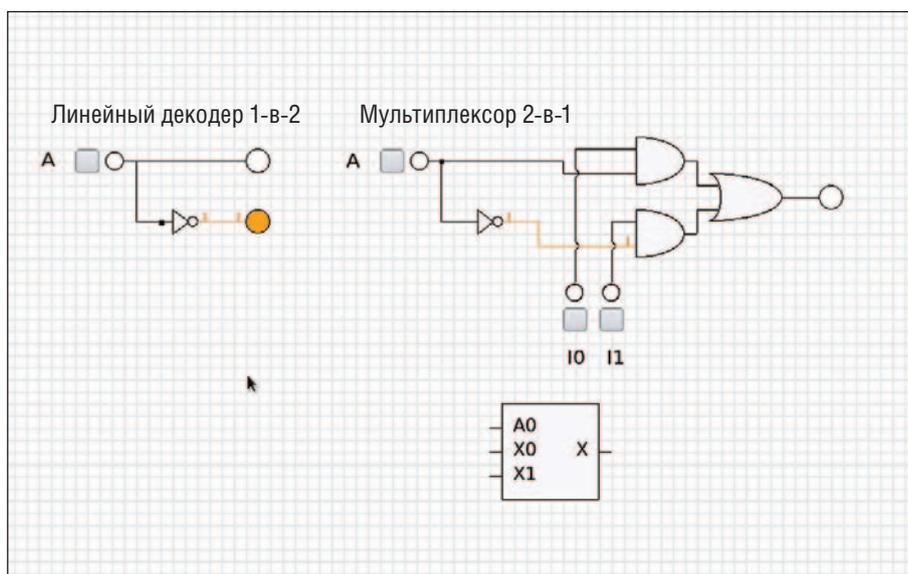
| A | D0 | D1 |
|---|----|----|
| 0 | 1  | 0  |
| 1 | 0  | 1  |

### Мультиплексоры

Мультиплексор подобен декодеру, но способен на большее. Вместо выбора между двумя линиями передачи данных, он использует адресную линию для выбора между двумя входами, и выбранная величина (**0** или **1**) будет перенаправлена на единственную линию передачи данных.

Его таблица истинности приведена ниже, а его схему соединений вы также можете увидеть на рис. 1. На этой схеме вы видите, что мультиплексор — это просто расширенный декодер.

| I1 | I0 | A | D |
|----|----|---|---|
| 0  | 0  | 0 | 0 |
| 0  | 0  | 1 | 0 |
| 0  | 1  | 0 | 0 |
| 0  | 1  | 1 | 1 |
| 1  | 0  | 0 | 1 |
| 1  | 0  | 1 | 0 |
| 1  | 1  | 0 | 1 |
| 1  | 1  | 1 | 1 |



» Рис. 1. На этой схеме вы видите линейный декодер 1-в-2 слева и мультиплексор 2-в-1 справа. Условное обозначение ниже — это встроенное представление KTechlab, где X0 и X1 — эквиваленты I0 и I1.

Декодеры и мультиплексоры полезны, поскольку позволяют выбирать между разными опциями. Полный сумматор, который мы сделали в прошлый раз, будет частью арифметико-логического устройства (АЛУ) компьютера. В настоящем компьютере АЛУ тоже выполняет много других операций, таких, как вычитание, умножение, логические AND, OR и NOT – и именно мультиплексоры и декодеры позволяют нам делать выбор между разными операциями.

## Больше логики

Чтобы это продемонстрировать, мы скомбинируем наш полный сумматор с прошлого раза с другими логическими схемами (**AND**, **OR** и **XOR**), и используем мультиплексор для указания операций, которые требуется выполнить.

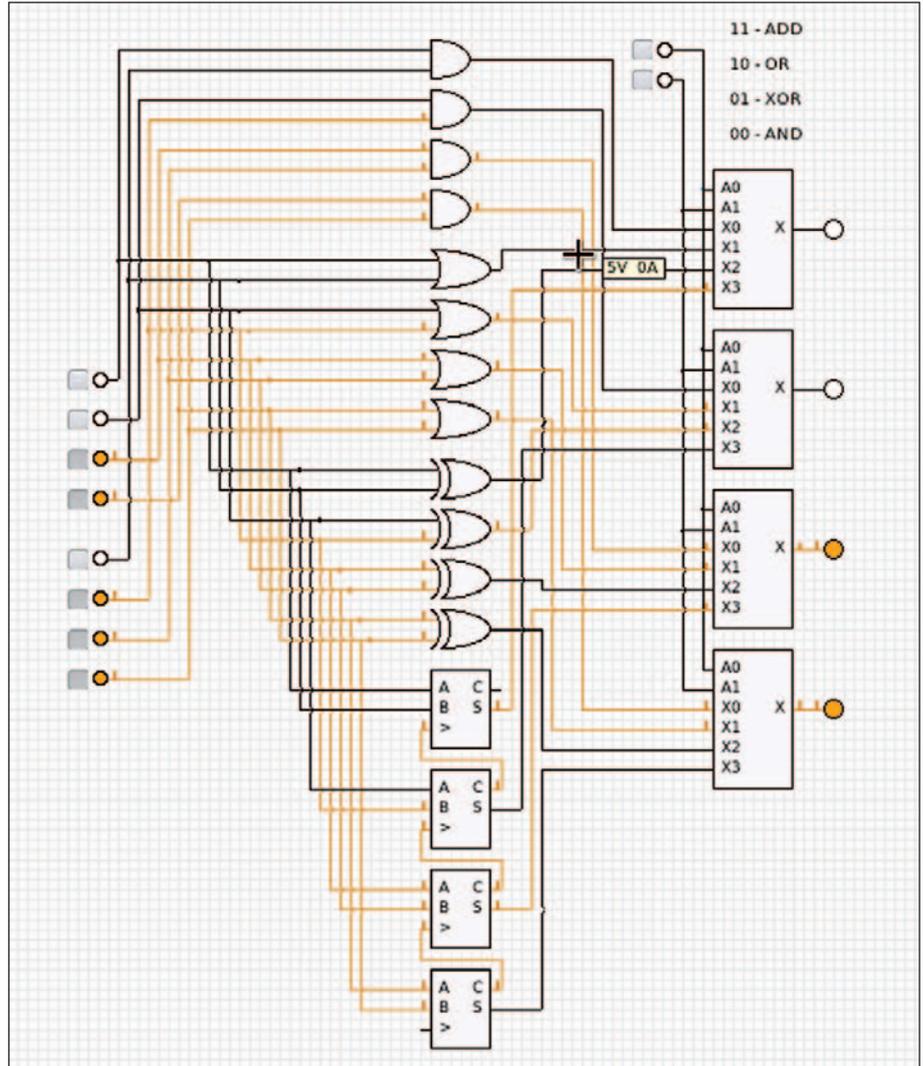
Первое, что нужно сделать – создать логическую схему. Это просто, поскольку, чтобы выполнить **AND** для двух двоичных чисел, все, что вам нужно сделать – это соединить каждый из первых битов с входом **AND**, вторые биты – с другим, третьи – с другим, и так далее. То же верно и для операций **OR** и **XOR**. Пока что можете оставить выходы несоединенными.

Присмотревшись внимательнее, вы сможете увидеть все это на рис. 2. Однако помните, что *KTechlab* не очень хорошо работает со сложными схемами, подобными этой, и будьте внимательны, рисуя соединения – проще всего будет перевести это в режим Manual, выбрав нужную опцию под значком с красной ручкой.

Сделав это, создайте четыре мультиплексора, настроив длину адреса каждого на два. Это можно сделать, щелкнув по мультиплексору, затем заглянув в появившуюся панель справа (Item Editor). Получится мультиплексор с четырьмя входами, между которыми вы сможете выбирать, создавая разные комбинации двух адресных линий.

Далее, соедините первый вход **AND**, вход **OR**, вход **XOR** и сумматор с первым мультиплексором, второй вход – со вторым мультиплексором, и так далее. Затем соедините логические выводы с соединением мультиплексора, отмеченным **x**, и, наконец, присоедините еще два логических ввода: один – ко всем **A0**, и другой – ко всем **A1** на мультиплексорах.

Все это показано на рис. 2, хотя вы можете также загрузить схему с диска – **ALU.circuit** – и рассмотреть ее хорошенько. Важная подсказка: все



» Рис. 2. АЛУ, использующее мультиплексоры для обеспечения выбора между операциями.

должно быть выровнено, потому что начнется большая путаница, если биты одного и другого входов пойдут в разном порядке, а порядок выходных битов тоже окажется другим!

Рассмотрев схему поближе, вы увидите рядом с адресными входами список значений, на которые они должны быть настроены для выбора разных операций. В настоящем компьютере это именуются opcodes (сокращение от «коды операций»). Их нужно передать процессору, чтобы заста-

вить его выполнить определенное действие. рядом с кодами находятся трехбуквенные описания функций – это мнемоника, для простоты запоминания и человеко-читаемого описания того, что делают разные коды. Если вы когда-нибудь писали на языке ассемблера, то это как раз и есть мнемонические значения, которые компиляторы ассемблера затем превращают в двоичные эквиваленты.

Вот таким образом и создается схема, которую можно проинструктировать на выполнение всяких операций. Однако на данный момент ввод в эту схему осуществляется непосредственно нами – мы вручную настраиваем логический вход, а выход являет собой логические выводы, которые включаются и выключаются.

А если мы хотим, чтобы выход не просто исчезал, но превращался во вход следующего устройства? Или если мы хотим запрограммировать входы заранее?

## Обратная связь

Чтобы это сделать, нам нужна память, и именно это станет предметом остальной части нашей

## Другие полезные операции

Мы использовали **AND**, **OR** и **XOR** как примеры операций для расширения нашего сумматора, потому что эти схемы просты в создании. Но если вы работаете с числами без знака, то логический сдвиг влево или вправо может оказаться более естественным выбором для расширения нашего сумматора.

Как и предполагает название, логический сдвиг влево сдвигает все биты на одну позицию влево, а логический сдвиг вправо – на одну позицию

вправо. Бит, который выпадает, потерян навсегда, а свободное место, появившееся в начале, занимает **0**.

Если число является беззнаковой переменной, эти операции имеют эффект умножения (сдвиг влево) или деления (сдвиг вправо) числа на два. Это должно быть очевидно, поскольку каждая позиция имеет вес в два раза больший, чем позиция справа от нее.

статьи. Но прежде чем перейти к схемам памяти, начнем с рассмотрения понятия обратной связи, которая лежит в основе схем памяти.

Возьмем обычный вентиль **OR**. Напомним его таблицу истинности:

| I0 | I1 | O |
|----|----|---|
| 0  | 0  | 0 |
| 1  | 0  | 1 |
| 0  | 1  | 1 |
| 1  | 1  | 1 |

Вентиль **OR** будет гарантированно вести себя именно так, если только... Что произойдет, если подключить выход ко входу? Что если создать сигнал обратной связи?

Посмотрите на схему на рис. 3 слева. Здесь вы видите, как мы ответили выход и соединили его напрямую с одним из входов. Оставшийся вход по-прежнему присоединен к нашему логическому вентилю.

Пока оставшийся вход – это **0**, все устройство остается в этом состоянии. Однако стоит вам переключить вход на **1**, как устройство заблокируется. Выход будет постоянно передавать **1** обратно на вход, и из этого состояния будет не выйти. Образно выражаясь, вентиль заперт. Попробуйте создать эту схему, чтобы понять, как она работает.

## Триггер задержки

Очевидно, что этой простой схемы обратной связи недостаточно для применения в качестве ячейки памяти – задав это состояние, вы не сможете переустановить его! Если вы записали туда порцию данных, а затем решили записать еще, то вам не повезло. Есть другие схемы, которые используют тот же принцип обратной связи, но без такого ограничения. Одна из них известна под названием «триггер D-типа» или триггер задержки, и мы его используем, чтобы продемонстрировать, как в компьютерах работает память.

Взгляните на схему в правой части рис. 3. Это промежуточный шаг между нашей простой схемой обратной связи и триггером задержки. Он известен под названием «закрытый триггер S-R» или триггер пересчета. Если в этой схеме оба входа установлены в **нуль (zero)**, выходы остаются

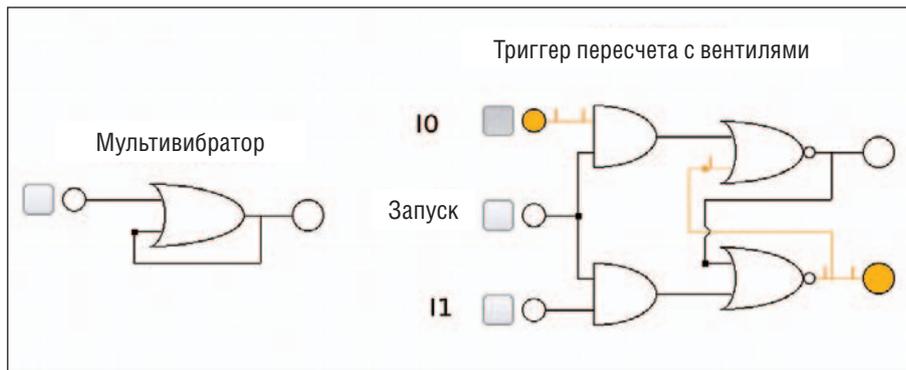


Рис. 3. Схема обратной связи слева демонстрирует основной принцип цифровой памяти, а триггер пересчета справа показывает, как получить больший контроль над схемой.

блокированными – иными словами, они помнят свое предыдущее состояние. Если верхний вход установлен в **1**, то нижний выход будет **1**; а если нижний вход равен **1**, верхний выход равен **1**.

Идея этой схемы в том, что два выхода всегда должны быть в противоположных состояниях. Однако так будет не всегда: если одновременно за-

## «Идея — что два выхода всегда в противоположных состояниях.»

дать на оба входа **1**, оба выхода станут равны **0**. Эту ситуацию можно обойти, создав триггер задержки, который также дает нам большую степень контроля за операциями схемы. Один из них можно увидеть на рис. 4.

Самая важная функция в этой схеме на текущий момент в том, что, соединяя входы в единое управление и инвертируя его, мы обеспечиваем такую ситуацию, когда схема не может оказаться в нештатном положении. Второе, что стоит отметить, это использование вентилей **AND** для «перекрытия» схемы. Это обеспечивает второй вход, который должен быть переключен на **1**, чтобы активировать схему. Когда она активируется, верхний выход будет всегда отражать вход, но когда она отключена, схема останется заблокированной, что бы вы ни делали со входом.

Этот ввод часто именуют не **Запуск [Enable]**, а **Такты [Clock]**, и далее будет понятно, почему.

Если не обращать внимания на нижний выход и сфокусироваться только на верхнем, вы увидите, что у нас получилась 1-битная ячейка памяти. Вы пишете **1** или **0**, активируя тактовый сигнал, и затем задаете вход, чтобы он имел любую величину, которую вы хотите сохранить. Когда вы отключите такты, ячейка сохранит эту величину, и вы сможете считать ее с верхнего выхода.

Начиная с настоящего момента и далее, когда нам понадобятся триггеры задержки, мы будем использовать встроенное представление *KTechLab*. Вы найдете его в разделе *Integrated Circuits*, где оно именуется “D Flip-Flop [мультивибратор]”. Они почти идентичны, но все же обязательно прочитайте раздел *Запускающий перепад сигнала*, чтобы понять разницу между триггером задержки и D Flip-Flop.

## Применим результаты

Мы можем легко интегрировать эту память в созданное нами АЛУ, заменив индикаторы вывода триггерами задержки – по одному на каждый переключатель. Тогда память сможет запомнить результаты последнего вычисления. Вроде и не слишком полезно, но зато дает возможность соединить разные операции в нашем АЛУ, используя т.н. регистр-сумматор.

Чтобы это сделать, придется подвергнуть ранее созданное нами АЛУ некоторому редизайну. Его результаты показаны на рис. 5. Принципы почти те же: между входами и выходами по-прежнему

## Срабатывание по перепаду сигнала

В созданном нами триггере задержки вход действует, только когда переключатель **enable** установлен на **1**. Это прекрасно, но в компьютерах и в цифровой электронике в целом удобны схемы, которые активируются за более короткое время, то есть пока вход **enable** переходит из одного состояния в другое. Это позволяет создавать намного более чувствительные схемы, с более тонким контролем над ними. Когда схемы, подобные триггеру задержки, запускаются во время переброса

входа **enable** из **0** в **1**, это называется уже не триггером, а мультивибратором. Мы говорим о «срабатывании по перепаду сигнала [edge triggering]».

Быть может, это звучит чуть ли не мистически, но если вы представите себе график прямоугольной волны, все станет ясно. Когда линия высоко, то вход или выход равны **1**; когда она низко, это **0**. В случае с триггером задержки выходы повторяют входы все время, пока **enable** высоко. В случае с мультивибратором, он будет реагировать только на одно реб-

ро прямоугольного графика, образованного тогда, когда вход **enable** переходит в состояние **1**.

Это не просто интересно, поскольку при работе со встроенным D Flip-Flop от *KTechLab* вам нужно обеспечить включение и выключение тактового сигнала, чтобы он действовал. Он срабатывает при переходе от высокого уровня к низкому, так что вам нужно установить переключатель на **1** и снова на **0**, чтобы активировать мультивибратор и заставить его сохранить новую величину.

существуют мультиплексоры, чтобы осуществлять выбор между разными операциями, и имеет набор кодов операций на выбор. Однако многое и изменилось. Самое важное, что мы интегрировали триггеры задержки в вывод, чтобы схема могла запомнить предыдущий результат. Это и есть регистр.

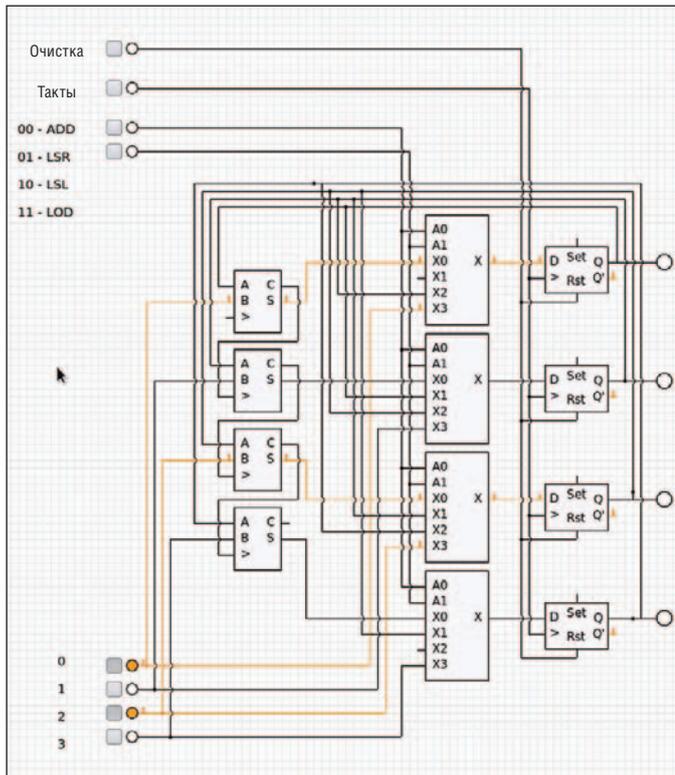
Суть регистра в том, что это как бы электронный блокнот – место, куда заносятся числа для дальнейшей обработки. Это ускоряет сложные операции, поскольку избавляет процессор от необходимости записывать промежуточные результаты в память и затем загружать их снова, чтобы выполнить операцию.

Это становится возможным благодаря использованию выхода регистра в качестве одного из входов, так что в оставшейся части схемы содержится только один набор переключателей входа, а остальные входы поступают с выхода регистра.

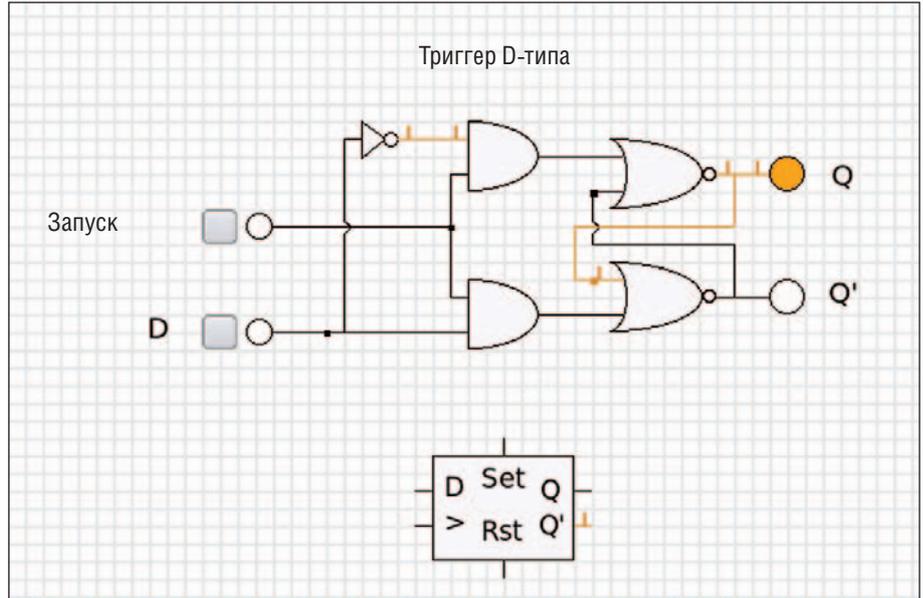
## Новые команды

Имеется также несколько новых инструкций-команд, включая логический сдвиг влево и вправо (реализуемый простым соединением выхода регистра непосредственно со входами мультиплексоров, смещенными на одну позицию влево или вправо).

Также имеется команда **LOD**, или **load**, которая вводит первое число для обработки в регистр. И наконец, вы также можете обратить внимание, что мы соединили Rst на каждом триггере с переключателем, чтобы можно было переключить состояние регистра в полный **0** одним нажатием.



► Рис. 5. Наше законченное АЛУ демонстрирует, как можно интегрировать схему памяти в процессор.



► Рис. 4. Триггер D-типа работает как 1-битная схема памяти, решая проблемы с закрытым фиксатором SR. Под основной схемой располагается презентация встроенного D FlipFlop от KTechlab.

Переключатель **clock** является эквивалентом **enable** в ранее созданном нами триггере задержки, но обязательно прочтите врезку «Срабатывание по перепаду сигнала», чтобы понять, как это работает.

Создав такую схему, вы можете загрузить число в регистр, ввести другое число на остальные входные переключатели, выбрать операцию переключателями opcode, а затем включить переключатель **clock** и выполнить операцию над содержимым регистра – если это команда **ADD**, то и над содержимым переключателей ввода (опе-

рации **LSL** и **LSR** обрабатывают только содержимое регистра). Если вы затем захотите выполнить другую операцию с результатом, то и такое будет возможно.

Вот и все, что мы собирались рассмотреть в статье этого месяца. Теперь у вас должно быть встроенное АЛУ с возможностью выполнения разных операций, как определено в специальных opcode, способных соединить эти операции благодаря вводу регистра.

Нам предстоит еще долгий путь. В следующем выпуске мы рассмотрим, как интегрировать в процессор ОЗУ, соединив его с устройством управления, и ввести в ОЗУ программу, чтобы процессор выполнил всю серию команд. **LXF**

## Больше, больше, больше

С триггерами задержки можно сделать еще и не то – возможно, самым важным будет возможность создать разновидность оперативной памяти (ОЗУ). Суть ОЗУ в соединении серии 1-битных схем хранения, в нашем случае – триггеров задержки, таким образом, чтобы вы при желании могли читать или записывать определенные биты, находящиеся на хранении.

Большинство ОЗУ не обладает структурой, позволяющей читать и записывать отдельные биты: они структурированы скорее для чтения и записи групп битов, размер которых в процессоре известен как длина слова. Большинство современных компьютеров работают с длиной слова, которая равна 32 или 64 битам.

В нашем процессоре длиной слова выбрана величина 4 бита. Решив создать собственную схему ОЗУ, вы, очевидно, сможете это сделать, руководствуясь накопленными знаниями.

Вам нужно будет обеспечить возможность адресоваться к каждому слову в памяти – отличным решением станет декодер – и обдумать, как обращаться к каждому слову в памяти без необходимости записывать значения вводимых битов, а также изобрести способ читать отдельное слово. Здесь мог бы пригодиться вентиль **AND**. И, наконец, вам понадобятся некоторые входные биты, чтобы устанавливать значения внутри памяти, и выходные биты, чтобы проверять содержимое слова, к которому вы обращаетесь.

# openSUSE LiveCD: Устанавливаем!

**Алексей Федорчук** обнаружил нетривиальные возможности установки openSUSE в Live-режиме: при желании работа с ним проходит не так уж и бесследно.

**Б**ольшинство современных дистрибутивов Linux распространяется в двух основных вариантах: в виде образов оптических дисков, служащих исключительно для установки системы, и в виде образов LiveCD/DVD, предназначенных как для знакомства с ней, так и для последующей инсталляции. Не исключение тут и openSUSE, официальный набор образов которой включает:

- » полный установочный DVD;
- » небольшой образ для установки по Сети;
- » Gnome-Live и KDE-Live CD, различающиеся используемыми рабочими средами.

О последних и пойдет речь в настоящей статье.

## Вступление

Среди записных линуксоидов, вне зависимости от используемого дистрибутива, существует несколько пренебрежительное отношение к установке системы с «живых» дисков: обычно считается, что этот метод предназначен для совершенно зеленых пользователей, устанавливающих свой первый в жизни Linux. Пользователям же с опытом должно применять альтернативные установочные носители.

Это мнение имеет под собой основания: в большинстве случаев при установке с LiveCD возможности пользователя вмешаться в этот процесс минимальны или напрочь отсутствуют. Результатом же такой установки является точная копия образа «живого» диска со всеми его умолчальными настройками и заранее предопределенным набором приложений. А поскольку в отношении

последних предпочтения составителей LiveCD наверняка не на 100% совпадают с предпочтениями пользователя «со стажем», в дальнейшем ему придется затратить немало времени на индивидуализацию своей системы.

Однако инсталляция openSUSE с LiveCD оказывается исключением из общего правила. И предоставляет не меньше возможностей для индивидуализации системы, нежели установка с полного DVD или «сетевого» диска. А в отношении настроек – даже больше. Как это оказывается возможным, и будет предметом настоящей статьи.

Как уже было сказано, официально в рамках проекта распространяется два варианта LiveCD – с KDE и Gnome в качестве рабочих сред, каждый в сборках для 32-битной и 64-битной архитектур. В силу личных предпочтений автора дальнейшее изложение проводится на примере KDE-LiveCD – но пользователи Gnome при установке своего любимого десктопа вполне могут применить те же приемы.

## Live-среда: запуск

Работа в Live-режиме, будь то знакомство с системой или ее установка, начинается с загрузки с соответствующего носителя. В ходе ее весьма желательно выбрать русский язык интерфейса. Правда, для Live-среды это мало что даст, ибо на 700 МБ вместить полную поддержку даже основных языков, как это имеет место быть на DVD, довольно сложно, а дожидаться предпочтения русскому от интернационального дистрибутива было бы опрометчиво. Но в случае последующей

инсталляции русская локализация будет унаследована установленной системой – хотя в ходе ее придется мириться со смесью нижегородского с оксфордским (рис. 1). Зато в дальнейшем для окончательной русификации потребуется куда меньше телодвижений.

Из меню загрузчика следует, что можно либо загрузить Live-среду, из которой будет доступна опция установки, либо непосредственно приступить к инсталляции. В данный момент нас интересует как раз первый вариант. Почему он является предпочтительным, станет ясным из дальнейшего изложения.

Никаких дополнительных опций, вроде настройки сети, Live-вариант загрузки пока не предусматривает – этим можно будет заняться уже непосредственно в «живом» режиме. Так что нажимаем Enter на первом пункте главного меню и через некоторое время видим рабочий стол KDE (рис. 2).

Первая наша цель – ознакомиться с возможностями Live-режима. Однако делать это лучше в комфортной обстановке, что для меня, например, подразумевает в первую очередь шрифты, подходящие для глаз, для кого-то – иные параметры внешнего вида. Чем мы для начала и займемся. Впрочем, те, кого внешний вид среды по умолчанию устраивает, могут спокойно пропустить следующий раздел.

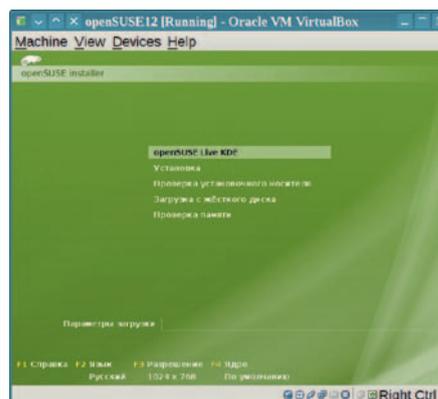
И еще важное предупреждение: знакомство с LiveCD – занятие довольно медленное и печальное. Ибо привод компактвс нынче не самое быстрое действующее устройство хранения данных,

## Из неофициальных источников...

Официальные LiveCD в отношении версий ядра, рабочих сред и приложений точно соответствуют установочному DVD на момент выхода текущего релиза. Однако существуют и периодические «верстовые [Milestones]» сборки, предназначенные для тестирования релиза будущего. По своему наполнению они идентичны официальным, однако версии этого наполнения повышаются «от столба к столбу». В промежутках же между «верстовыми столбами» с периодичностью примерно раз в неделю собираются «саженные столбики» – снапшоты текущего состояния подготавливаемого

релиза. Разумеется, за стабильность «верстовых» и особенно «саженных» сборок никто не ручается, и использование их для «рабочих» инсталляций не рекомендуется.

Кроме официальных LiveCD, существует большое количество неофициальных их вариантов. Например, для KDE это версии с «чистым» KDE 4.X (то есть в оригинальном оформлении проекта KDE) и с ностальгическим KDE 3.5.10. Сборки LiveCD с прочими рабочими средами – XFce, LXDE, Enlightenment – также имеют статус неофициальных.



» Рис. 1. Меню установщика являют собой языковую смесь.



► Рис. 2. Капли росы на зеленом листке...

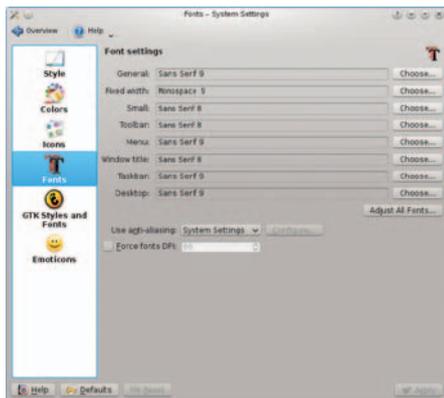
а кэширование его содержимого в оперативную память (даже если ее более чем вдоволь) в openSUSE не предусмотрено.

## Подготовка Live-режима

Дабы привести рабочий стол Live-среды в соответствие если и не со своими эстетическими идеалами, то хотя бы с физическими возможностями восприятия, отправляемся в стартовое меню главной управляющей панели, где выбираем пункт Configure desktop (я предупреждал, что с Великим и Могучим в Live-среде будет напряженка) и по открытии окна настройки параметров щелкаем по иконке Applications Arrearance.

В развернувшейся панели слева выбираем пункт Fonts, а справа жмем на кнопку Adjust all fonts (рис. 3).

Теперь отмечаем «птицами» боковые Font и Size и выбираем подходящие гарнитуру и кегль, сверяясь с образцом в нижней части окошка. Выбрав подходящий вариант, в панели шрифтов включаем режим anti-aliasing'a в соответствии со своим визуальным восприятием. Каковой, впрочем, будет получен только впоследствии, после перезапуска сеанса (но ни в коем случае не системы – это убьет все сделанные настройки). Однако перед этим я подгоняю управляющую панель к размеру, воспринимаемому моими глазами. Для чего щелкаю на ней правой кнопкой мыши, в контекстном меню выбираю пункт Panel Options, а затем – Panel Settings. После этого, ухватившись мышью



► Рис. 3. Настройке поддаются шрифты практически всех компонентов системы.

за кнопку Height, ташу ее вверх до получения удовлетворительного результата.

Вот теперь в первом приближении дело с настройкой можно считать законченным – остается только перезапустить сеанс KDE и авторизоваться заново, введя в качестве логина значение linux, а поле для ввода пароля оставив пустым. После

чего рабочий стол KDE загрузится снова – но уже с применением всех сделанных ранее настроек.

## Настройка окружения root

Итак, мы привели внешний вид рабочего стола в приемлемое состояние. Однако не следует этим ограничиваться: нам предстоит еще некоторые действия, которые надо будет выполнить от имени администратора, а на его окружение пользовательские настройки не распространяются. Да и установка системы тоже будет происходить в окружении суперпользователя. Так что последний штрих в подготовке к дальнейшей работе –

## «Мы привели внешний вид рабочего стола в приемлемое состояние.»

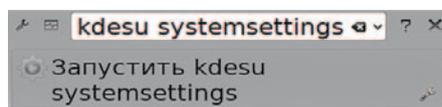
это настройка «административного окружения». Для чего следует запустить ту же самую программу конфигурирования рабочего стола, но уже от лица суперпользователя. Самый простой способ сделать это – с помощью комбинации клавиш Alt+F2 вызвать командную строку мини-терминала, в которой и надлежит ввести

```
kdesu systemsettings
```

где **kdesu** – команда для получения временных, на одну операцию, прав суперпользователя, а **systemsettings** – команда для запуска программы установки параметров рабочего стола, что и показано на рис. 4.

После этого мы получаем доступ к настройкам окружения рабочего стола суперпользователя, где следует проделать все действия по настройке шрифтов – точно так же, как это было описано в предыдущем разделе.

Предвижу резонное замечание: зачем возиться с настройками окружения пользователя и администратора, если все эти действия будут иметь силу только для текущего Live-сеанса? Столь же



► Рис. 4. Окно мини-терминала.

резонно возражу. Во-первых, не такие уж это сложные действия, чтобы пренебречь комфортом в ходе знакомства с LiveCD и установки. А во-вторых, и главных: не пропадет ваш скорбный труд по настройке пользовательского и административного окружения. Почему? Пусть это пока остается Военной Тайной.

## Преамбула к установке

На знакомстве с Live-средой я останавливаться не буду. Замечу только, что это самая обычная среда KDE, с набором типовых ее приложений – достаточно обширным, так что начинающему пользователю есть где порезвиться. Однако занятие это наскучит ему достаточно скоро – не в последнюю очередь и потому, что приложения не блистают быстродействием в условиях «живого» режима. И тут пользователю захочется посмотреть на них во всей красе – в инсталлированном виде. Наступает психологический момент для установки системы.

Установка openSUSE – дело не шести секунд. Конечно, в Live-режиме это время можно скрасить рядом приятных и полезных занятий. Так, те, кто еще не наигрался в игрушки, имеют все условия, чтобы резаться, скажем, в *Reversi* или раскладывать пасьянсы, каковых немного больше чем вдоволь. Люди же серьезные могут почитать материалы официального сайта проекта и документацию, в том числе и на русском языке. Благо, для этой цели в Live-режиме имеется целых два браузера.

Конечно, тут хорошо бы иметь и соединение с Интернетом. Если провайдер обеспечивает DHCP-подключение, все просто: сеть волшебным образом поднимется сама собой. Однако в данном варианте не фатально будет и любое другое подключение: в нашем распоряжении есть рабочий *Network Manager*, который, при всех его недостатках, позволит настроить и VPN-, и DSL-, и Wi-Fi-соединение. Ибо нет таких настроек, которые не могли бы выполнить большевики-линуксоиды, и сеть, тем или иным способом, будет поднята. И никаких препятствий к повышению своего образовательного уровня не будет.

Так что возможность занять время установкой делами разной степени полезности – это первый довод в пользу установки системы из Live-среды, а не методом «лобового напора» – из меню загрузчика.

Однако чтение материалов, как я уже сказал, занятие для серьезных людей. Люди же несерьезные, вроде автора этих строк, предпочтут провести время установки за непринужденными беседами, например, в Джуйке – <http://juick.com/>. И на первый взгляд их ожидает облом: в Live-среде ни малейшего Jabber-клиента не найти и следов.

Ну, это решается легко: если Jabber-клиента в системе нет, его следует установить. На вопрос «как?» ответить опять же легко: либо с помощью консольной системы *zypper*, либо посредством модуля управления программами универсальной системы YaST2 в графическом режиме. Оба

## Борьба за независимость

Дабы при удалении пакетов вместе с ними удалялись и их ставшие ненужными зависимости (если они больше нигде не задействованы), при использовании *zypper*'а следует отредактировать файл `/etc/zypp/zypp.conf`, а именно: снять символ комментария со строки

```
# solver.cleandepsOnRemove = false
```

и заменить значение `false` на `true`.

При использовании же модуля управления пакетами *YaST2* тот же эффект достигается включением в меню Параметры пункта Удалять ставшие ненужными зависимости.

Чтобы пакеты при установке тянули за собой только обязательные зависимости, не трогая так называемых рекомендованных, в файле `/etc/zypp/zypp.conf` уберите символ комментария со строки

```
# solver.onlyRequires = false
```

и замените значение `false` на `true`.

Та же цель в модуле управления пакетами *YaST2* достигается включением в меню Параметры пункта Игнорировать рекомендованные пакеты для уже установленных пакетов.

Однако торопиться с отключением установки рекомендованных пакетов не следует. Ибо в число оных входят и языково-зависимые пакеты, о которых говорилось ранее. Так что, вместо того чтобы устанавливать их по пакетно, достаточно при отключенном игнорировании рекомендованных пакетов выполнить операцию тотального обновления системы командой `zypper up` или через *YaST2*. Результатом будет полная русификация системы.

способа – предмет отдельной темы. Здесь лишь отмечу, что во втором-то случае и пригодились нам настройки «административного окружения», ибо *YaST2* запускается от лица суперпользователя.

Возможность поговорить с приятными собеседниками online – это второй довод в пользу инсталляции openSUSE из Live-среды. Правда, те самые серьезные люди поставят это в упрек: мол, не стоит ради пустопорожнего треп городить огород с установкой дополнительных приложений. На что у меня есть два возражения:

1) Треп в Джуйке никогда не бывает совершенно пустым, и кроме эмоциональной разрядки приносит и практическую пользу – в виде ответов на вопросы, часто и нетривиальные, в реальном времени.

2) Как это ни парадоксально, но приложения, установленные в Live-среде, сохраняются в инсталлированной системе; почему – опять же будет предметом отдельного разговора.

*Kopete* может оказаться далеко не единственным кандидатом на установку. Так, не помешает установить в Live-среде и пакеты русификации, такие, как `kde4-l10n-ru`, `kde4-l10n-ru-data` и `kde4-l10n-ru-doc` для русификации KDE, `libreoffice-l10n-ru` для русификации LibreOffice. Впрочем, полностью русифицировать систему можно и иным способом (см. врезку). Прочие дополнительные пакеты каждый выбирает в меру своих предпочтений.

Наконец, самое интересное: пакеты можно не только устанавливать в Live-среду, но и удалять из нее – и после инсталляции на диск их не будет! Здесь я «отдельно, с большим наслаждением» удаляю немало того, что полагаю лишним на «живом» диске, в частности, всю штатную мультимедиа, заменяя ее *Mplayer*'ом – но и это дело личных предпочтений.

Выходит, Live-среда дает ничуть не меньше возможностей для индивидуализации системы, нежели выборочная установка с DVD или по сети. И даже больше: потому что никто не в силах запретить подключение, наряду со штатными, также и сторонних репозиторий, в том числе содержащих так называемые не вполне свободные программы (типа мультимедиа-кодексов: Flash-

плейера etc.), которые при других методах установки системы приходится доустанавливать впоследствии.

Таким образом, третий довод в пользу установки из Live-режима – безграничные возможности по индивидуализации системы. Причем реализуются все эти возможности, что называется, малой кровью: пользователь может полностью избавиться от заботы о зависимостях как при установке пакетов, так и при их удалении.

Последнее представляется мне особенно ценным: сколько я ни занимался установкой с индивидуальным выбором пакетов как в openSUSE, так и в «более иных» дистрибутивах, и как бы ни старался вычеркнуть из предлагаемого разблюдовника компоненты, ненужные лично мне, все равно половина из них пролазила в инсталлированную систему в качестве чьих-то зависимостей.

При установке же с openSUSE LiveCD от всего лишнего можно избавиться радикально. Потому как предвзято можно должным образом настроить *YaST* или отредактировать конфигурационный файл *zypper*'а, в зависимости от того, что используется для удаления и установки пакетов.

Четвертый довод в пользу установки в Live-режиме по сравнению с прямой инсталляцией – возможность выполнять ее в визуальном приятном окружении, о чем говорилось в предыдущем разделе.

Есть и пятый довод, но и он пока останется Маленьким Секретом, который я раскрою под занавес.

## Установка

И вот настал решительный момент щелкнуть мышью по иконке Install на предмет заняться установкой системы. Она начинается с панели приглашения к оной (рис. 5).

После приглашения можно видеть отличительные особенности инсталляции в Live-режиме:

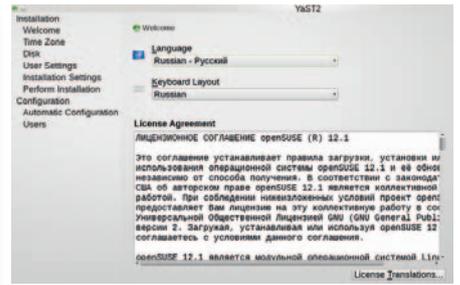


Рис. 5. Как полагается, все начинается с лицензии.

- » Нет пункта выбора режимов – то есть режим обновления установленной системы не предусмотрен.
- » Нет возможности отключить автоматическую настройку после инсталляции.
- » Нельзя включить использование диска Add-on.

Впрочем, ни об одной из этих опций особо жалеть не стоит – все эти вопросы решаемы другими методами. Так что сразу после приглашения переходим к определению часового пояса, где можно также скорректировать время и включить синхронизацию с серверами NTP (рис. 6).

Стадия выбора рабочего стола при Live-установке пропускается. Что и естественно: выбор этот делается в тот момент, когда диск с соответствующей средой (KDE или Gnome) ставится на зачку.

Так что следующим номером нашей программы будет разметка диска – возможности установщика openSUSE в этой области поистине необъятны, так что здесь мы на них задерживаться не будем, это должно быть предметом отдельного разговора. Далее создается учетная запись обычного пользователя, после чего выводится итоговая панель Live Installation Settings (то, что в русском переводе интерфейса типовой установки обзвали Параметрами установки) – рис. 7.

Но не ищите здесь секции индивидуального выбора пакетов: ее здесь нет. Да и не нужна она, ибо все необходимые пакеты мы имели возможность установить «вживе» еще до запуска инсталлятора.

Теперь по нажатию кнопки Install будет запрошено подтверждение этого судьбоносного решения. А дальше процесс разметки диска, создания и монтирования файловых систем, а также собственно установки пойдет сам по себе.

Пока он идет, ответим на вопрос, как нам удалось устанавливать и удалять пакеты, да так, что

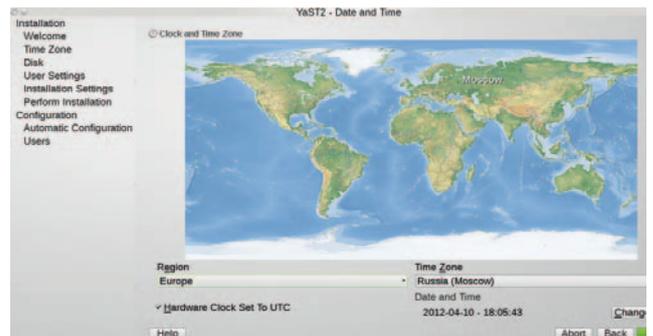
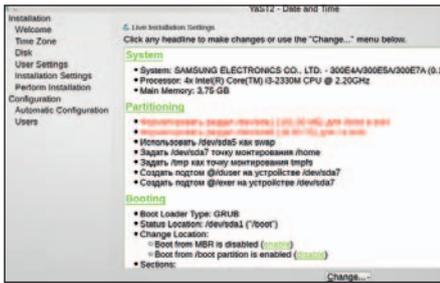


Рис. 6. Московское время – 18:05.



► Рис. 7. Меняйте, что вашей душе угодно.

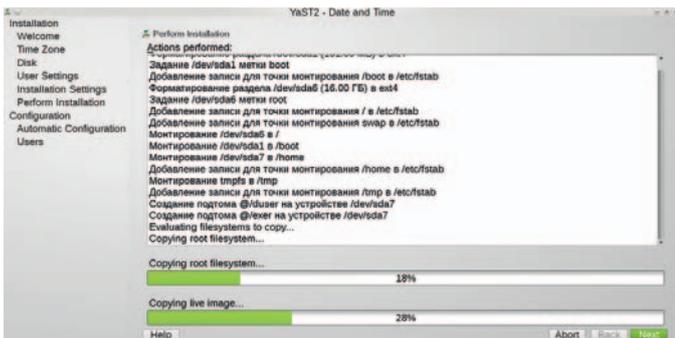
сделанные в Live-среде изменения сохраняются в инсталлированной системе. Хотя, казалось бы, после перезагрузки они должны были бы бесследно исчезнуть. Для чегождемся в окне инсталлятора окончания разметки диска и расправы с файловыми системами. После чего текущим действием будет одно-единственное – копирование корневой файловой системы [Copying root filesystem], рис. 8.

Вот вам и разгадка Военной Тайны. Ибо где расположена корневая файловая система Live-среды? Правильно, в оперативной памяти. А куда инкорпорируются исполняемые файлы, библиотеки и прочие компоненты установленных в ходе Live-сеанса пакетов? В корневую файловую систему. А откуда изымаются компоненты пакетов удаляемых? И где отражаются изменения, выполненные в общесистемных конфигах? Опять-таки, все это модификации корневой файловой системы – точнее, ее образа в оперативной памяти.

Так что в процессе установки с LiveCD не происходит ни развертывания образов метапакетов, ни по пакетной распаковки индивидуально выбранных пакетов. По сути, дело сводится к переносу текущего слежка оперативной памяти на целевой носитель. И потому на нем по завершении установки все изменения, сделанные в Live-среде до запуска инсталлятора, сохраняются в неприкосновенности. В этом и заключается сила описанного метода – насколько мне известно, не имеющего аналогов в других дистрибутивах Linux.

## Итоги установки

Вот теперь можно и на Джуйк переключиться. Одна беда: за время наблюдений за процессом установки и размышлений о его сути установка-то и закончилась. Что неудивительно: ведь ко-



► Рис. 8. Процесс копирования пошел.

пирование образа из оперативной памяти на современный винчестер – дело достаточно быстрое, куда быстрее, чем распаковка пакетов и распределение их компонентов по ветвям файлового древа. Так что очень скоро мы увидим предложение перезагрузить машину – немедленно или когда угодно позднее.

Торопиться с перезагрузкой мы не будем. Ибо пора раскрыть Маленький Секрет пятого довода в пользу установки из Live-режима: это возможность сохранения пользовательских настроек рабочей среды. Причем за настройки учетной записи администратора можно не волноваться: поскольку каталог `/root`, где они упокоились, лежит на корневой файловой системе, все конфигурационные файлы из него будут скопированы в соответствующее место на винчестере.

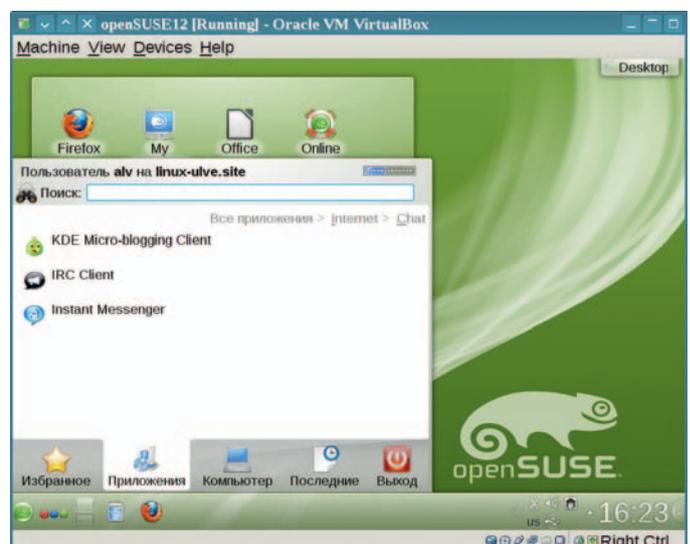
А вот настройки пользователя среды были сделаны для временного пользователя с именем `linux`, и его домашний каталог `/home/linux`, существующий в Live-режиме, при перезагрузке будет уничтожен.

Однако никто не запрещает нам скопировать конфиги уходящего в небытие пользователя `linux`, например, на флешку, а затем перенести их в установленную систему. Или сразу поместить их в `/mnt_point/home/username`, где `mnt_point` – точка монтирования для раздела на винчестере (не следует забывать, что по окончании установки все действовавшие во время нее файловые системы размонтируются), а `username` – учетное имя пользователя, чья учетная запись была создана во время установки. Нужно будет только потом изменить их владельца и проверить права доступа.

Файлы, подлежащие копированию, – это в первую очередь конфиги KDE (`/home/linux/.kde`) и Kopete (`/home/linux/.kde4/share/config/kopeterc`), а возможно, и других установленных в Live-среде программ.

Вот теперь можно и перезагрузиться. Установка в Live-режиме не предполагает отказа от автоматического конфигурирования системы. Каково и происходит сразу после ее рестарта. И завершается появлением умолчального рабочего стола KDE.

► Рис. 9. Наши труды не пропали даром: все настройки сохранены.



Впрочем, вид его не совсем умолчальный. Бледный взгляд на главное меню показывает, что оно стало русифицированным, утратило пункты, соответствующие пакетам-отказникам, и, напротив, в его закоулках мы найдем приложения, установленные ранее в Live-сеансе. А шрифты и главная панель сохраняют тот вид, который мы придали им перед установкой (рис. 9).

## Закключение

Из всего сказанного выше можно сделать вывод, что установка с LiveCD – отнюдь не обязательно прерогатива совсем уж начинающих линуксоидов. Конечно, для них такая инсталляция As Is обеспечивает максимально быстрое развертывание системы, содержащей необходимый для начала работы минимум приложений. Но если затратить некоторое время на настройку и наращивание возможностей Live-среды, то систему эту можно сделать и актуальной, и индивидуализированной. Причем существенно более простыми методами, чем при ручном выборе пакетов при установке с DVD или NET-диска.

Правда, этот метод можно рекомендовать только пользователям с достаточным опытом работы в других дистрибутивах. Ибо устойчивые предпочтения в отношении рабочего окружения и прикладного софта у них наверняка уже сложились, и что им надо получить в итоге – они сами знают. Так что для достижения неизменно превосходного результата им потребуется только знакомство со специфической для openSUSE системой управления пакетами.

Следует, впрочем, оговорить, что способ этот подходит только тем, кто отдает предпочтение средам KDE 4 или Gnome 3, потому что официальных LiveCD с другими рабочими столами просто нет. Что же до неофициальных – они обычно выходят с некоторой (а иногда и значительной) задержкой относительно текущего релиза. Хотя и эта проблема в принципе решается – но с существенно большими затратами времени и сил. Так что любителям *Xfce*, *LXDE* или, тем более, оконных менеджеров проше прибегнуть к установке с полного DVD или диска для сетевой инсталляции. **LXF**



Что за штука...

# SecureBoot

**Маянк Шарма**, в роли адвоката дьявола, раскрывает сумбурную переделку Microsoft процесса загрузки вашей системы.

**В** Минуточку! Я уже слышал о SecureBoot. Это же одно из решений безопасности в новом Windows 8, так?

**О** И да, и нет. SecureBoot – это механизм улучшения безопасности системы посредством привязки к процессу загрузки. По факту, это функция спецификации UEFI. С Windows 8 ее связывает только то, что, по требованию Microsoft, она должна быть включена во всех машинах, где стоит сертифицированная «восьмерка».

**В** UEFI? Уверен, что и про это мне уже доводилось слышать.

**О** Ну да, мы обсуждали его на наших страницах, в LXF153. UEFI, то есть Unified Extensive Firmware Interface, приходит на замену интерфейсу BIOS, который мы использовали все эти годы. UEFI был создан в Intel еще в 1990-е, а теперь им занимается консорциум крупных технологических компаний, включая AMD, Microsoft, ARM и другие.

**В** Ясно, вернемся к SecureBoot. Как именно она способствует безопасности?

**О** В ранних версиях UEFI для проверки целостности драйверов и программ использовались электронные подписи. Этот механизм, основанный на проверке цифровых сертификатов, использовал известную технологию Public Key Infrastructure (PKI). А в UEFI версии 2.2, вышедшей в 2008, поддержка электронных подписей расширена и распространена также на загрузчики ОС.

**В** А как это поможет предотвратить атаки злоумышленников?

**О** На ПК с включенной SecureBoot обновленный процесс загрузки Windows 8 гарантирует, что все элементы прошивки и их обновления безопасны, и что система защиты компьютера не повреждена.

**В** Хорошо, но где тут гарантия, что не повреждена сама прошивка?

**О** Не позволяя вредоносному коду загружаться при старте системы, SecureBoot пропустит только проверенный и подтвержденный код, например, такой, как загрузчик операционной системы. Все загрузчики и драйверы должны иметь подпись, соответствующую одному из проверенных сертификатов. Это защитит вас от вирусов сектора загрузки, а также руткитов, маскирующихся под драйверы.

**В** Но как UEFI сможет проверить загрузчик операционной системы?

**О** Вскоре после старта системы, прежде чем передать бразды правления загрузчику ОС, SecureBoot проверит подпись всех кодов прошивки подключаемых устройств, таких как сетевые карты, запоминающие устройства или видеокарты. Затем она сверит все эти подписи с базой данных, и только тогда модулю будет разрешено запуститься.

на них наклейку «Совместим с Windows 8». Но для установки самой системы по-прежнему достаточно только соответствия технических требований.

**В** То есть, на любом компьютере с наклейкой «Совместим с Windows 8» будет UEFI с включенной функцией SecureBoot?

**О** Да, именно это я и имел в виду, вы правильно меня поняли.

**В** Ух ты. То есть мамин новый ноутбук на Windows 8 будет безопасен как никогда прямо из коробки? Пожалуй, дам ей его поиграть, пока она привыкает к Ubuntu.

**О** Погодите радоваться. Теперь получается, что установка новой операционной системы приравнивается к установке неаутентифицированного кода, поскольку на ней отсутствует подпись, а значит, невозможно будет устанавливать дистрибутивы.

**В** Что? Вы хотите сказать — я не смогу установить Linux на свой нетбук с сертифицированной Windows 8?

**О** По крайней мере, не сразу. Имею в виду, что наша несчастная прошивка UEFI опирается на свою базу данных подписей, чтобы отличить вредоносный руткит от нормальной ОС. Исходя из этого, UEFI не позволит Linux совершать какие-либо действия с вашим компьютером, если только

**О** На данный момент, только два популярных дистрибутива – Fedora и Ubuntu – раскрыли свои стратегии по работе с SecureBoot. Fedora подпишет свой загрузчик ключом Microsoft, чтобы ее можно было гарантировано установить на любой стандартный ПК с Windows 8. В Ubuntu выбрали подход посложнее. Дистрибутив будет выходить с собственной подписью, которую пользователям надо будет добавить в базу данных на своем компьютере.

**В** А почему используются разные подходы? И который лучше?

**О** На самом деле, и с тем, и с другим связан ряд технических и юридических сложностей. И, между прочим, Фонду свободного ПО [FSF, Free Software Foundation] не нравится ни тот, ни другой. Первый – потому что пользователям Fedora придется зависеть от подписи Microsoft и от самой Microsoft, как от благодетеля. Со вторым такой проблемы нет, зато добавляется куча всяких сложностей, поскольку пользователям придется копаться в настройках UEFI, чтобы подтвердить достоверность подписи Canonical.

**В** А как насчет малых дистрибутивов, которые вне мейнстрима?

**О** Прочие Linux-дистрибутивы пока что можно установить на оборудование с сертифицированным Windows 8 только при условии отключения SecureBoot.

**В** О, так ее, оказывается, можно просто взять да отключить?

**О** Да. В разъяснении своей сертификационной программы для производителей оборудования, Microsoft разрешает им применять функцию отключения SecureBoot.

**В** Уфф! Я уж было испугался. А выходит, не все так плохо.

**О** Это еще не все. Во-первых, это не значит, что предусмотрен стандартный способ ее отключения. Речь шла о том, чтобы разрешить производителям оборудования использовать механизмы отключения данной функции по собственному усмотрению. А стало быть, каждый разработает свой индивидуальный способ, и это ни в коей мере не упростит существование дистрибутивам.

**В** Ну и пусть это будет непросто, но пользователи все равно смогут по желанию отключить SecureBoot, так?

**О** Да, но в идеале этого не должно требоваться. Как нетрудно понять, сам механизм SecureBoot всех полностью устраивает. Уж где-где, а в дистрибутивах Linux безопасность всегда была превыше всего, а отключив SecureBoot, вы тут же лишаетесь преимуществ, которые она предоставляет. Кроме того, отключение будет возможно только на ПК с архитектурой x86. А в машинах на основе ARM, сертифицированных Windows 8, Microsoft отключать SecureBoot запрещает. **LXF**

## «SecureBoot пропустит только проверенный и подтвержденный код.»

**В** И где же хранятся эти замечательные подписи?

**О** Подписи хранятся в двух разных базах данных, а также в самой прошивке UEFI. Они представляют собой два списка – «Разрешенные» и «Запрещенные», и в соответствии с ними процесс загрузки может быть продолжен или завершен, если в подписи распознан вредоносный код.

**В** Хитро придумано. Но откуда мне эти подписи взять?

**О** Это не ваша головная боль. Согласно созданной Microsoft новой программе Сертификации Windows, на всех новых компьютерах с предустановленной системой Windows 8 или совместимых с ней ПК, ноутбуках или устройствах на основе ARM, типа планшетов, база подписей будет по умолчанию. Более того, производители оборудования должны будут подтвердить, что функция SecureBoot в их машинах действительно включена.

**В** Выходит, я не смогу установить Windows 8 на ноутбук, который я заказал?

**О** Конечно, можете. Microsoft просто попросил производителей активировать SecureBoot на всех новых компьютерах, прежде чем лепить

у него нет подписей для данного конкретного дистрибутива.

**В** А нельзя из одного дистрибутива подписать другой?

**О** Можно, конечно. Но сделать это будет непросто. Для запуска, даже в случае, когда дистрибутив подписывает себя сам, все равно потребуются подписи производителя прошивки. Это ресурсоемкая задача невыполнимая для широкого спектра дистрибутивов. Есть и куда большая проблема. В нынешнем виде, подписать ядро для SecureBoot – это еще полдела, поскольку даже в этом случае ядро не будет загружать неподписанные модули.

**В** И даже если мой любимый дистрибутив подпишет ядро, я не смогу загрузить дополнительные модули?

**О** При включенной SecureBoot – нет. Как объяснил в своем блоге разработчик Red Hat Мэтью Гарретт [Mathew Garrett], это значит, что придется распрощаться с любым проектом, требующим загрузки обязательных модулей.

**В** И как же дистрибутивы ухитряются с этим справиться?



# 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.

## Комедия ошибок

**В** командной строке всегда гораздо проще сделать что-то неверно, чем верно. Это находит свое отражение в разнообразии сообщений об ошибках, встречающихся в Linux. От некоторых хочется убежать и спрятаться, другие можно понять, только имея под рукой исходный код – но есть и очень яркие. Они мне нравятся.

Вот несколько примеров и программы, которыми они принадлежат:

```
Cowardly refusing to create an empty archive
[Трусливо отказываюсь создать пустой архив] – tar
```

```
I refuse to debug myself! [Отказываюсь
отлаживать сам себя!] – gdb
```

```
Didn't think there would be that many child
processes... Exiting [Не ждал, что дочерних
процессов так много... Ужжу] – foomatic
```

```
Null message body; hope that's ok ()
[Сообщение пусто; надеюсь, все ОК] – mail
Manufacturer is guessed because of the
orange forum embargo [Изготовитель ука-
зан по догадке из-за эмбарго апельсино-
вого форума] – cdrecord
```

## Строки в комплекте

Интересный набор сообщений выудит из системных двоичных файлов команда **strings**:

```
strings -f -n 20 /bin/* /usr/bin/*
```

Учтите, что команда выдает почти миллион строк – естественно, без контекста, и неясно, какая ошибка вызвала эту бурную реакцию.

Итак, в этом месяце у меня для вас два задания. Во-первых, пришлите на мою почту самое интригующее сообщение об ошибке, которое вы видели (для Linux, пожалуйста, не копайте в прошлом), и, во-вторых, придумайте сообщение об ошибке, которое, на ваш взгляд, должно быть в Linux, а его нет. Лучшие будут опубликованы.

А если вам понравилась эта тема, обширная коллекция сообщений имеется на сайте [http://neil.franklin.ch/Jokes\\_and\\_Fun/Canon\\_Error\\_Messages](http://neil.franklin.ch/Jokes_and_Fun/Canon_Error_Messages) (по большей части из эпохи до Linux).

chris.linuxformat@gmail.com

## Эзотерическое системное администрирование из причудливых заворотов кишок серверной



## Туманные тарифы

Выбор облачных сервисов на AWS становится таким же сложным, как и заказ кофе в Старбаксе.

**Н**а заре эпохи Amazon Web Services все было просто: создаете несколько экземпляров компьютеров, подключаете к ним жесткие диски – и за пару секунд готов почасовой тариф для вашей конфигурации. Увы, все меняется. Теперь, если вы не защитили докторскую диссертацию по теме «Геодизическая полнота бесконечно дифференцируемых Римановых пространств», расчет тарифа сведет вас с ума.

Но не бойтесь – помощь рядом. У Amazon на <http://calculator.s3.amazonaws.com/calc5.html> есть «Простой калькулятор месячного тарифа». Узнав, сколько вам нужно экземпляров компьютеров по требованию и сколько зарезервированных экземпляров, и какие из 12 доступных типов экземпляров, сколько эластичных IP-адресов, и какой объем места на диске (и какая его часть будет «избыточной») могут вам понадобиться, сколько запросов **PUT** и **GET** вы выполните и сколько

VPN-соединений установите, а также... и... (в общем, список довольно длинный, но, уверен, вы поняли идею), калькулятор посчитает стоимость. На рисунке внизу показана только одна из 17 вкладок по сервисам Amazon. Это какое-то новое значение слова «простой», раньше я о нем не знал.

У Rackspace тоже есть калькулятор стоимости (<http://www.rackspace.co.uk/cloudhosting/learn-more/calculator>); он гораздо проще, прежде всего потому, что набор сервисов здесь гораздо беднее.

Допустим, я понимаю преимущества быстрой масштабируемости и почасовой тарификации, а также переноса капитальных затрат в операционные, но мне все же интересно: когда, наконец, кто-нибудь скажет, что расходами управлять проще, имея собственный дата-центр?

А что думаю я? Мне, пожалуйста, обезжиренный венте мокка фраппучино с 50 % содержанием кофеина, с собой.



Simple Monthly Calculator от Amazon.

## Бесплатные пробы

Попробовать web-сервисы Amazon можно легко (и бесплатно), особенно если у вас уже есть обычная учетная запись Amazon.

Amazon предлагает бесплатный годовой период с момента регистрации, в течение которого можно попробовать ключевые сервисы AWS. Однако если вы выйдете за пределы допустимых ресурсов или за 12 месяцев тестового периода, то услуги придется оплатить.

# Новое поколение средств защиты

## Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



## Назначение устройства

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

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

**Дополнительное устройство защиты для:**

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

Москва  
+7 (499)

271-49-54

Санкт-Петербург  
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. [www.linuxcenter.ru](http://www.linuxcenter.ru)

Linux  center

# Итак, вы хотите стать сисадмином?

Вторая часть серии, которая превратит вас из новичка в звезду системного администрирования. Поговорим о настройке сети.

**В** прошлом месяце мы говорили о способах удаленного подключения к серверам, которые вы администрируете. Мы обсудили такие утилиты, как *ssh* и *VNC*. На сей раз мы поговорим об основах настройки сети на сервере. Вы научитесь:

- » Задавать IP-адрес, маску подсети и другие параметры сетевых интерфейсов
- » Задавать маршрут по умолчанию
- » Задавать статические маршруты к заданным сетям
- » Задавать расположение DNS-серверов
- » Настраивать Linux для работы в качестве маршрутизатора

Во всех статьях этой серии мы работаем с CentOS 6.2. Если вы хотите последовать нашему примеру (что я настоятельно советую сделать), установите CentOS (например, в виртуальную машину), как описано на предыдущем уроке. Итак, начнем...

Серверы (по определению) существуют не изолированно, а подключаются к сетям. На главном рисунке показана типичная ситуация, в которой может оказаться сервер. Основные персонажи нашей истории (показанные на рисунке) таковы:

**A:** Главный герой нашей истории. Это компьютер с Linux, который мы настраиваем. Он находится в сети 10.1.3.0/24, одной из внутренних сетей компании.

**B:** Компьютер во внутренней сети 10.1.2.0/24.

**C:** Маршрутизатор, который связывает нашу внутреннюю сеть с корпоративной магистралью, сетью 10.1.1.0/24. Как и остальные маршрутизаторы на рисунке, он может быть и специализированным устройством от изготовителей вроде Cisco, и компьютером с Linux. На его примере мы рассмотрим настройку маршрутизатора Linux.

**D:** Маршрутизатор, связывающий внутреннюю сеть с корпоративной магистралью.

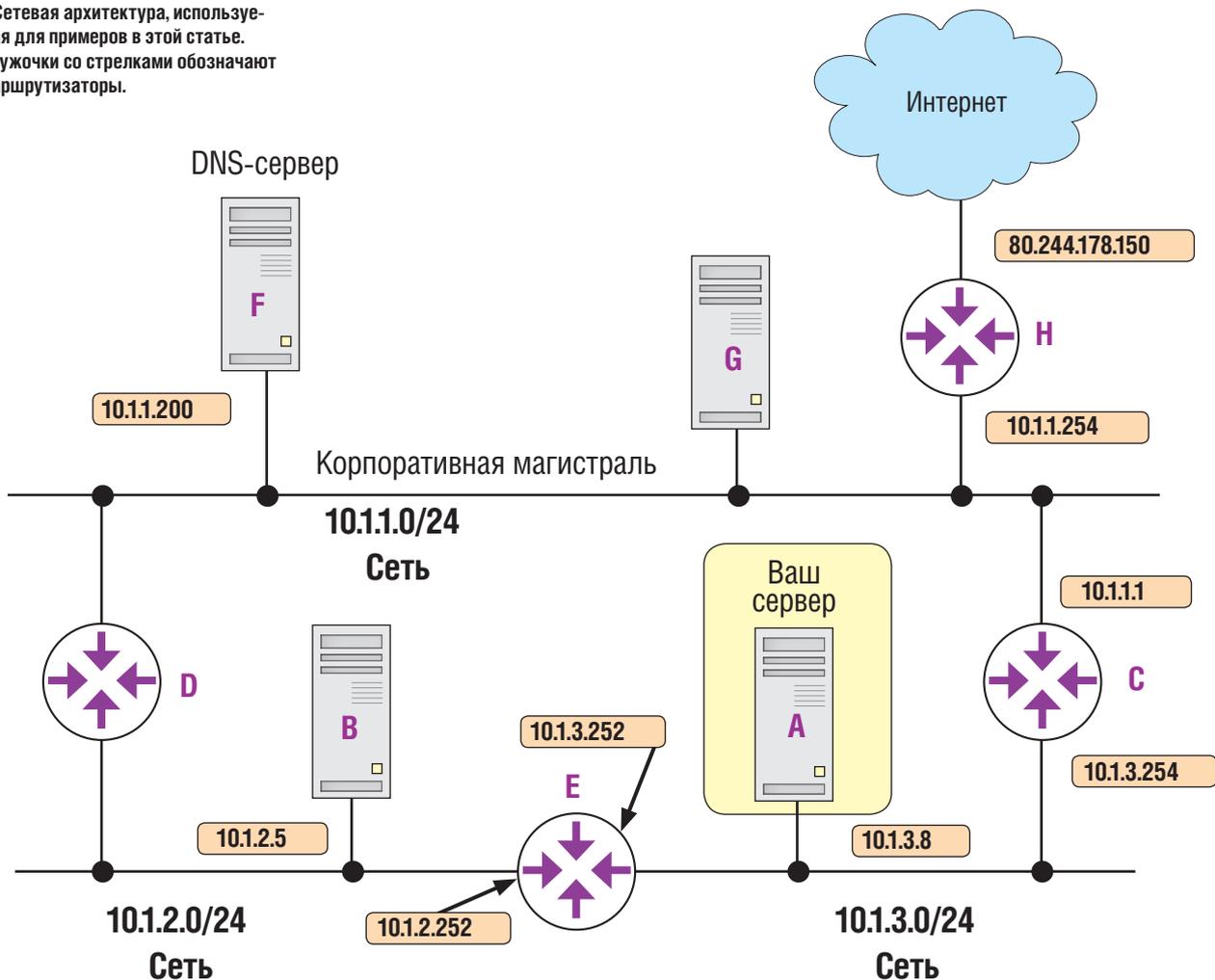
**E:** Маршрутизатор, связывающий две внутренние сети.

**F:** DNS-сервер компании (собственные DNS-серверы есть не у всех компаний, но у этой есть).

**G:** Другой компьютер на корпоративной магистрали.

**H:** Маршрутизатор, связывающий всю компанию с Интернетом. Его внешний IP-адрес – единственный доступный извне адрес на рисунке. Все остальные находятся в диапазоне 10.0.0.0/8, выделенном только для внутреннего частного использования.

» Сетевая архитектура, используемая для примеров в этой статье. Кружочки со стрелками обозначают маршрутизаторы.



Основная часть настройки сети – задание IP-адреса и маски подсети сетевого интерфейса нашего сервера. В большинстве дистрибутивов Linux это можно сделать в графических утилитах, но мы предполагаем, что на сервере нет графического рабочего стола. Поэтому посмотрим, как изменить соответствующие конфигурационные файлы. Все это обычные текстовые файлы, и сложностью они не отличаются.

В файле `/etc/sysconfig/network` задаются некоторые базовые системные настройки, то есть те, что применяются ко всем сетевым интерфейсам сервера, а не только к одному. Файл обычно очень короткий, нечто вроде этого:

```
NETWORKING=yes
HOSTNAME=centos62.example.com
```

Настройки для отдельных сетевых интерфейсов, в данном случае для `eth0`, задаются в файлах с именами типа `/etc/sysconfig/network-scripts/ifcfg-eth0`. Содержимое файла в CentOS 6.2 зависит от настроек сети, заданных во время установки системы, но в нем скорее всего будет следующая строка:

```
NM_CONTROLLED="yes"
```

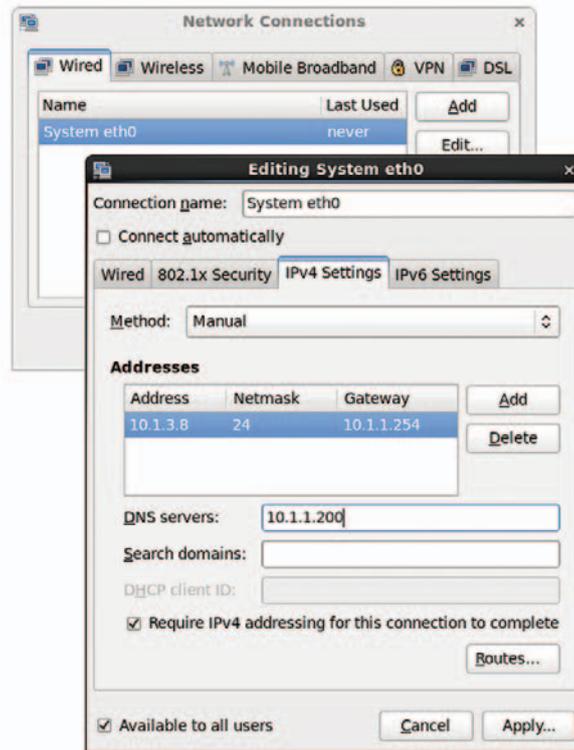
Она означает, что интерфейс находится под управлением *NetworkManager* – утилиты из Red Hat (перенятая другими дистрибутивами), которая пытается поддержать активное сетевое соединение для компьютеров, которые часто перемещаются между различными сетями, с минимальным вмешательством пользователя или вообще без него. Обнаружив проводное Ethernet-соединение, она активирует его, а если сделать это не удастся, то она подключится к сети Wi-Fi, к которой вы подключались ранее. Эта утилита весьма удобна для ноутбуков, но для серверов не годится. Поэтому мы избавимся от *NetworkManager* и настроим все вручную.

Сперва остановим *NetworkManager* и отменим его запуск во время загрузки:

```
# service NetworkManager stop
# chkconfig NetworkManager off
```

Прощай, *NetworkManager*. Далее откроем файл `ifcfgeth0` и зададим нужные параметры. Если вы следуете за нами, помните, что IP-адрес, который вы установите, должен соответствовать вашей сети и скорее всего не совпадет с указанным здесь. Вот простой пример:

```
DEVICE="eth0"
ONBOOT=yes
BOOTPROTO=none
IPADDR=10.1.3.8
NETMASK=255.255.255.0
IPV6INIT=no
```



Эта утилита Red Hat часто используется в графических утилитах настройки сетевых интерфейсов в различных дистрибутивах.

Будьте внимательны с синтаксисом файла. Он обрабатывается напрямую оболочкой, поэтому синтаксис должен быть верным. В частности, вокруг знака равенства не допускаются пробелы. Это следует соблюдать строго. Изменив этот файл, перезапустите сеть, чтобы изменения вошли в силу:

```
# system network restart
```

Эта команда остановит, а затем перезапустит все сетевые интерфейсы (в данном случае их два – интерфейс обратной петли и `eth0`). Если у вас несколько сетевых интерфейсов, то можно перезапустить только необходимые интерфейсы, следующим образом:

```
# ifdown eth0
# ifup eth0
```

Какой бы способ вы ни выбрали, теперь нужно проверить, что интерфейс активен – это делается так:

```
# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 08:00:27:6D:55:56
inet addr:10.1.3.8 Bcast:10.1.3.255 Mask:255.255.255.0
```

»

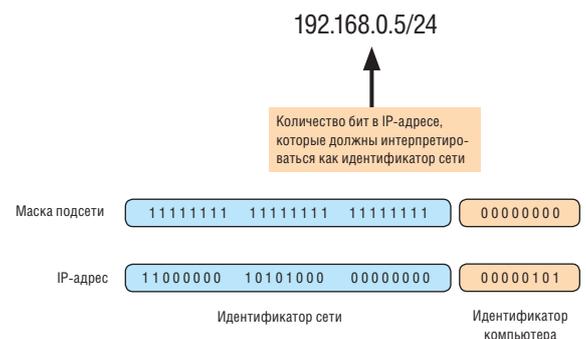
## Запись IP-адреса

Адрес IPv4 содержит 32 бита. При записи он разбивается на фрагменты по 8 бит, и они записываются в виде десятичных чисел (от 0 до 255) через точки, как показано на рисунке. Это называется десятичной записью через точку. Первая часть адреса – идентификатор сети; он идентифицирует сеть и используется при маршрутизации. Вторая часть – идентификатор компьютера; он идентифицирует конкретный компьютер (или, строго говоря, конкретное подключение) в данной сети.

Маска подсети в конкретной сети обозначает границу между идентификатором сети и идентификатором компьютера. Единицы в ней соответствуют идентификатору

сети, нули – идентификатору компьютера. К примеру, маска подсети `255.255.252.0`, в двоичном виде – 22 единицы и 10 нулей, означает, что первые 22 бита являются идентификатором сети.

Чаще всего вы увидите запись сетевой маски в виде суффикса. Например, в сети `10.1.3.0/24` идентификатором сети являются первые 24 бита адреса. В данном случае граница между идентификаторами сети и компьютера проходит по границе байта, но это не является обязательным требованием. В настоящее время в зависимости от контекста используются как более старая точечная десятичная запись, так и более новая запись с суффиксом.



IP-адрес состоит из идентификатора сети и идентификатора компьютера. Суффикс /24 означает место их разделения.

## Станьте экспертом

На map-страницах очень невразумительно описана общая картина настройки сети IP, поэтому я бы посоветовал вам книгу «Руководство администратора сети Linux» издательства О’Рэйли (третье издание). А если вы действительно хотите понять, как работают сети IP, прочтите «Иллюстрированный TCP/IP, том 1: Протоколы». Это второе издание книги Кевина Фолла [Kevin

Fall], основанной на классической работе покойного Ричарда Стивенса [Richard Stevens], которая, на мой взгляд, является лучшей книгой о TCP/IP.

Если вы хотите отделаться бесплатно, загрузите *Руководство и технический обзор TCP/IP* из Красной книги IBM с <http://www.redbooks.ibm.com/abstracts/gg243376.html?Open>.

```
UP BROADCAST RUNNING MULTICAST MTU:1500
Metric:1
Для краткости я удалил несколько строк вывода, но здесь важно то, что карте назначен адрес IPv4. Как видите, интерфейс активен и имеет полагающийся ему IP-адрес. Да, я знаю, что на map-странице ifconfig говорится, что такие адреса уже устарели. Но все продолжают ими пользоваться. Как написал Марк Твен, прочитав свой некролог в «Нью-Йорк Таймс»: «Слухи о моей смерти сильно преувеличены».
Еще один способ ощутить теплое, приятное и трепетное ощущение того, что интерфейс жив – попробовать попинговать его с другого компьютера. Если сервер Linux запущен на виртуальной машине, попробуйте попинговать его с хоста:
# ping -c1 10.1.3.8
PING 10.1.3.8 (10.1.3.8) 56(84) bytes of data:
64 bytes from 10.1.3.8: icmp_req=1 ttl=64 time=11.6 ms
--- 10.1.3.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 11.661/11.661/11.661/0.000 ms
```

### Финальные штрихи

Следующая задача – определить шлюз по умолчанию. На этот шлюз мы будем перенаправлять пакеты, предназначенные для компьютеров вне нашей сети, при условии, что более точного маршрута не задано. По сути, шлюз по умолчанию – последнее прибежище. На нашей схеме сети шлюз по умолчанию для сервера Linux – это нижнее подключение к маршрутизатору С. Шлюз по умолчанию можно указать в файле `ifcfg-eth0` следующим образом:

```
GATEWAY=10.1.3.254
Наконец, нужно сказать серверу, где находятся его DNS-серверы. Для этого нужно добавить в файл /etc/resolv.conf строку
nameserver 10.1.1.200
```

В данном случае мы пользуемся местным DNS-сервером на нашей корпоративной магистрали. Скорее всего этот сервер будет только кэшировать запросы, перенаправляя их на DNS-сервер более высокого уровня и запоминая последовательные запросы одних и тех же данных, которые можно сохранить в локальном кэше и затем быстро извлечь оттуда. Если у вас нет локального сервера DNS, укажите внешний.

Каким бы ни был основной сервер DNS, стоит указать вторичный (просто добавьте еще одну строку `nameserver` в `resolv.conf`), чтобы разрешение имен продолжало работать, если основной сервер выйдет из строя. Это один из немногих случаев, когда для смены конфигурации не нужно отправлять сигнал демону или что-то перезапускать. Измененный файл `resolv.conf` будет прочитан автоматически при выполнении следующего запроса DNS.

На этом базовая настройка сервера закончена. Однако кое-что можно улучшить. В данной конфигурации наш компьютер обратится к сети `10.1.2.0/24`, пройдя через шлюз по умолчанию С, а затем через маршрутизатор D. Но, конечно, есть и более короткий

путь – через маршрутизатор E. Чтобы сервер знал об этом маршруте, нужно добавить примерно такую строку в файл `/etc/sysconfig/network-scripts/routeeth0`:

```
10.1.2.0/24 via 10.1.3.252
Таблицу маршрутизации сервера можно посмотреть так:
$ netstat -nr
Kernel IP routing table
Destination Gateway Genmask Iface
10.1.3.0 0.0.0.0 255.255.255.0 eth0
10.1.2.0 10.1.3.252 255.255.255.0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 eth0
0.0.0.0 10.1.3.254 0.0.0.0 eth0
```

Я немного сократил вывод для экономии места. Таблица маршрутизации используется так: когда входящий пакет поступает на уровень обработчика протокола IP ядра, он сравнивается с каждым маршрутом в таблице по очереди. Сравнение выполняется путем побитового объединения по «И» IP-адреса назначения пакета со значением `Genmask` в таблице, а затем сравнением результата со значением `Destination`. Если они равны, пакет соответствует правилу. Из соответствующих правил пакетов выбирается самый конкретный маршрут (с самой длинным значением `Genmask`).

В первой строке таблицы задана сеть, напрямую подключенная к нашей (шлюз указан как `0.0.0.0`). Во второй строке определяется маршрут к сети `10.1.2.0` (это следует из строки, которую мы поместили в `route-eth0`). Третью строку добавил демон `Avahi`, реализующий архитектуру `Zerosconf` Apple; она не относится к нашей истории. Четвертая строка – маршрут по умолчанию. Любой пакет, для которого не нашлось более точного маршрута, будет соответствовать этому правилу, так как побитовое объединение по «И» любого адреса с `Genmask 0.0.0.0` будет соответствовать `Destination 0.0.0.0`.

### Настройка маршрутизатора

О сервере Linux, с его единственным сетевым интерфейсом, мы поговорили вдоволь. Поставим себе более амбициозную цель: настроим компьютер с Linux так, чтобы он играл роль маршрутизатора С на схеме. Это очень просто. У этого компьютера два сетевых интерфейса, `eth0` и `eth1`. Пусть `eth1` – верхний интерфейс с IP-адресом `10.1.1.1`, а `eth0` – нижний с IP-адресом `10.1.3.254`. Вам понадобится создать два конфигурационных файла `ifcfg-eth0` и `ifcfg-eth1`, указав в каждом соответствующий IP-адрес. Также не забудьте изменить строку `DEVICE=` в соответствии с именем интерфейса. Единственное, что осталось сделать для превращения компьютера в маршрутизатор – включить перенаправление IP-пакетов (под этим я понимаю прием пакета на один сетевой интерфейс и его перенаправление на другой). Ядро Linux не делает этого по умолчанию, но включить его легко – просто пропишите `1` в соответствующий параметр ядра таким образом:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
То же самое можно сделать и покороче:
# sysctl -w net.ipv4.ip_forward=1
Эта команда включит перенаправление IP-пакетов. Чтобы изменения стали постоянными, просто добавьте такую строку в /etc/sysctl.conf:
net.ipv4.ip_forward = 1
```

Готово! Вы только что настроили маршрутизатор! Если вы захотите попробовать это на `VMWare`, `VirtualBox` или другой технологии виртуализации, добавить еще один сетевой интерфейс в окнах настройки виртуальной машины можно парой щелчков мыши. На самом деле, при желании можно создать целую виртуальную сетевую инфраструктуру.

Через месяц мы поговорим об управлении хранилищами данных. Мы научимся добавлять диски, разбивать их на разделы, создавать файловые системы и многое другое. Увидимся! **LXF**

Наши эксперты помогут вам с любым приложением Linux!



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

## Растр умирает — да здравствует вектор!

Анна Каренина бросилась под поезд. Ну и что за сенсация? А то, что тогда в России большинство читателей ни разу не видели железной дороги.  
*Александр Митта*

Вы не поверите, но идею независимости от разрешения устройства в компьютерный мир занес Кнут с его системой разбиения параграфов на красивые абзацы TeX аж в 1978 году. Люди особо в это не вникли, ведь пиксели были прямо перед глазами и буквально их мозолили; и весь шум по поводу шрифтов на экране идет от неэффективности способов их маскировки. Что в общем бессмысленно. Человеческий глаз все равно замечает нестыковки: пиксели размером больше угловой минуты на самом удобном для чтения расстоянии гарантированно видны.

А ныне одна крупноблочная компания, эксплуатируя сумрачный корейский гений, токмо роста своих прибылей ради организовала рынок девайсов, где пикселей и не видно. Исчезли помехи для вашей фантазии и шрифтового десктопного безумия.

Чем это грозит? Придется с годами забыть про растр. Совсем. Как бы вы ни млели от любимого растрового логотипа вашего сайта, пора делать его векторным. Текст на экране теперь можно верстать как для печати, то есть не страдать насчет места букв на экранной решетке, а вволю применять те же TeXовские механизмы, но в реальном времени. Заодно это решит проблему утилизации возросшей процессорной мощи. Пора готовить GNU/Linux для десктопа будущего, даже если вы его пока не видели.  
[E.m.Baldin@inp.nsk.su](mailto:E.m.Baldin@inp.nsk.su)

## В этом месяце вы научитесь...



**Пресекать шпики** ..... 66  
**Джонатан Робертс** показывает, как применить ваш Raspberry Pi для SSH-туннелирования, чтобы никто не тырил у вас информацию.



**Придумывать пароль** ..... 70  
Переймите у **Джеймса Литтона** достойную стратегию назначения пароля, и ваши данные будут в безопасности. Пароль "password" он не советует.



**Избегать катастрофы** ..... 74  
Позвольте **Нейлу Ботвику** упрятать ваши данные в облаках — он продемонстрирует ряд полезных вариантов. Тогда гори все огнем, а данные и не охнут.



**Помогать юзерам** ..... 76  
**Майк Сондерс** научит вас составлять документацию, которая посрамит самые качественные образцы из былых. И это сумеют даже не программисты.



**Осваивать матрицу** ..... 80  
Серия уроков от **Ника Вейча** добралась до пристраивания к Arduino кнопок. Из монтажной платы выйдет неплохая клавиатура.



**Сканировать хосты** ..... 84  
**Игорь Штомпель** накопил графических интерфейсов к сетевому сканеру. Клавиатура — на покой, мыши — на старт.



**Облегчать нагрузку** ..... 88  
Вам нужно более легкое решение для виртуализации? **Джон Лэйн** представляет chroot на стероидах: Linux Containers.

## Пропустили номер?

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



# Raspberry Pi.

Пользуйтесь публичным Wi-Fi безопасно и одолейте цензоров: Джонатан Робертс рассказывает об SSH-туннелировании.



Наш эксперт

Супер-скрытный Джонатан Робертс копается в Linux с детских лет. Иными словами, не так уж долго...

Большинство из нас сталкивались с блокировкой интернет-соединения, не позволяющей делать нам то, что мы хотим — обычно в виде брандмауэра, ограничивающего доступ к сервисам, таким как SMTP или XMPP, или в виде фильтрации содержимого сайтов. Каким бы ни было ограничение, приятного здесь мало.

Большинство из нас также пользовались публичным Wi-Fi, и если у вас есть здравый смысл, то вы наверняка чувствовали себя не очень комфортно. В статье «Изучите взлом» из LXF154 было показано, как легко пользователи из той же самой сети могут шпионить за вами и даже влиять на ваши действия. С публичным

и шпионов! Raspberry Pi прекрасно подходит для этого. Так как он стоит недорого и работает бесшумно, его можно просто держать на полке, пока он не понадобится.

## Raspbian

Если вы следите за этой серией с самого начала, возможно, у вас еще остался исходный образ Debian, которым мы пользовались в первой статье. Однако Raspberry Pi Foundation не стоит на месте и недавно выпустила новую операционную систему на основе дистрибутива Raspbian.

Raspbian — новая версия Debian, оптимизированная для работы с Raspberry Pi. Она не только работает «прямо из коробки» и заметно быстрее (некоторые тесты показали прирост производительности от 4 до 40 %, в зависимости от рода деятельности), но и содержит удобную утилиту автоматизированной настройки, вызываемую при первом запуске Raspberry Pi. Она значительно упрощает многие задачи, решенные нами в предыдущих статьях, в том числе расширение раздела на весь диск, отключение overscan, включение SSH и изменение раскладки клавиатуры.

Советуем обновить вашу систему. Самый простой способ это сделать — загрузить последнюю версию образа с [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads), распаковать ее и записать образ на SD-карту командой `dd` так же, как мы делали это ранее.

Советуем очистить все содержимое диска, так как мы столкнулись с какими-то странными проблемами, когда этого не сделали — поскольку старые файлы настройки и программы сохранились. Воспользуйтесь следующей командой:

```
sudo dd if=/dev/zero of=/dev/mmcblk0 bs=1M
```

Здесь `mmcblk0` — устройство, соответствующее SD-карте. Перед запуском команды заранее создайте резервную копию, так как команда записывает нули на весь диск, делая все существующие данные невозможными.

## Порты

Прежде чем воспользоваться SSH-туннелем, объясним идею, лежащую в его основе.

При создании соединения между двумя компьютерами — для передачи web-документов, открытия защищенной оболочки или обмена файлами по NFS — компьютеру, иницирующему соединение, должны быть известны IP-адрес компьютера, к которому нужно подключиться, и номер порта сервиса (web, SSH, NFS и т.д.), к которому нужно обратиться.

Большинство из вас знают, что такое IP-адрес. Это число (например, **192.168.133.20**), которое является уникальным идентификатором компьютера в сети, точно так же, как почтовый адрес идентифицирует ваш дом. Но порты знакомы нам чуть хуже.

## Задаем порт

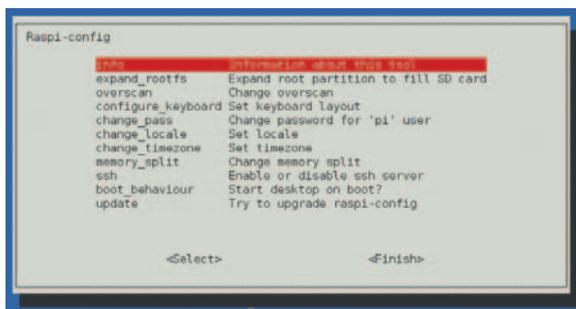
Так же как в одном доме обычно живет несколько человек, и каждый из них ждет свою почту, на каждом компьютере есть много разных сервисов, ожидающих подключения непосредственно к ним. В реальном мире мы адресуем почту конкретному человеку, положив ее в конверт с его именем, а в компьютерной

## «Raspbian — новая версия Debian, оптимизированная для работы на Raspberry Pi.»

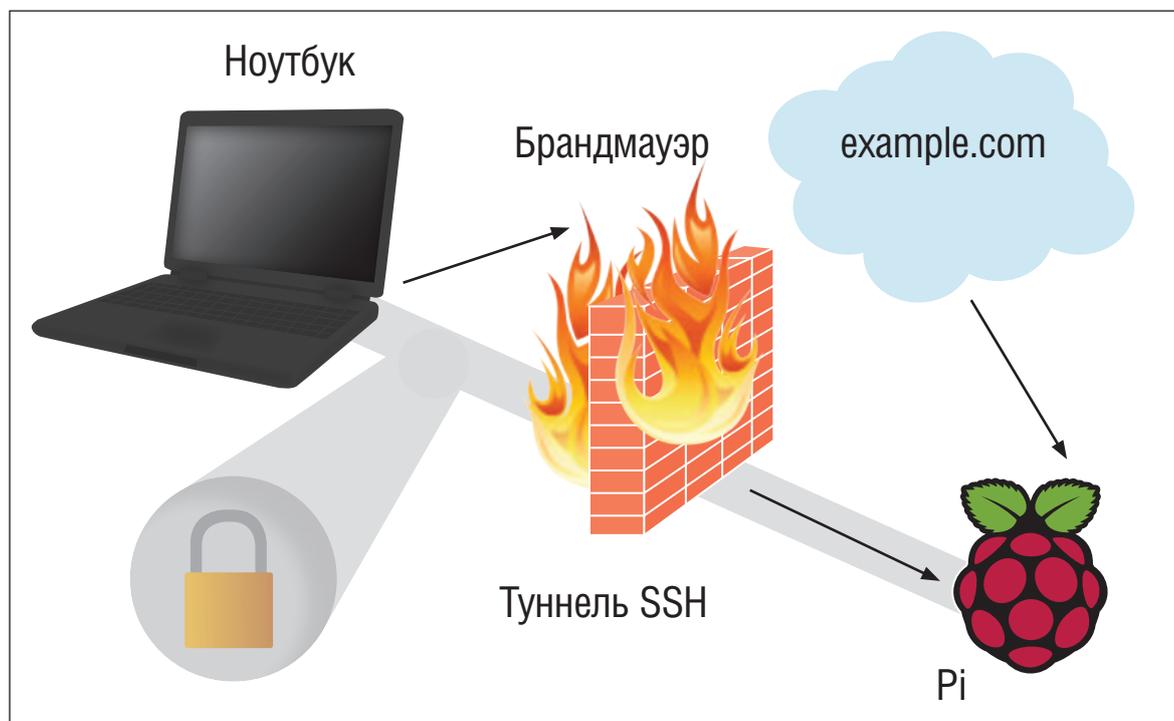
Wi-Fi вы не представляете, кто еще пользуется вашим подключением и что создатели подключения делают с вашим трафиком!

На этом уроке мы покажем вам, как создать SSH-туннель в Raspberry Pi. Эта технология поможет вам одолеть цензоров

» С новой утилитой `raspi-config` легко настроить Raspberry Pi. Просто запустите `sudo raspi-config`, и дело в шляпе.



# SSH-ТУННЕЛЬ



» SSH-туннель создает защищенное соединение между двумя компьютерами, с помощью которого можно обойти прокси и другие фильтры.

сети мы указываем конкретный порт, к которому намереваемся подключиться.

Каждый порт задается номером от 0 до 65535, и каждый сервис, предоставляемый компьютером, слушает соединения на одном из этих портов. Есть список портов по умолчанию, которые слушают все популярные сервисы. Этот список поддерживается IANA (Internet Assigned Numbers Authority – Администрация адресного пространства Интернета) и покрывает порты с 0 по 1023 (web-трафик проходит через порт 80, FTP – через порт 20, SSH – через порт 22 и т.д.).

Ничто не мешает вам настроить сервис на другой порт, и если вы знаете, на каком порту запущен сервис, можно указать клиенту подключаться к этому порту, а не к порту по умолчанию.

Хорошо, но какое отношение это имеет к SSH-туннелям?

На практике SSH-туннелирование сводится к искусному перенаправлению портов.

Лучше всего объяснить это на примере. Представьте, что у нас два компьютера: ноутбук в кафе (**laptop**), где доступ к Интернету ограничен прокси-сервером (программой, которая умеет фильтровать интернет-трафик) и нельзя зайти на сайт **example.com**, и **pi** – Raspberry Pi с неограниченным доступом в Интернет дома. У ноутбука есть доступ к **pi**.

## Туннелируем соединение

Чтобы открыть **example.com** с ноутбука, понадобится создать туннель, т.е. перенаправить наше подключение через **pi**. SSH-туннель такое позволяет.

Обычно SSH создает защищенное соединение между двумя компьютерами на порту 22. Но при использовании SSH-туннеля она перенаправит весь трафик с указанного порта **laptop** по зашифрованному соединению на порт 22 **pi** (или на любой другой указанный нами компьютер и порт).

У такого подхода два преимущества. Во-первых, как и при обычном SSH-соединении, трафик теперь направляется по защищенному соединению, то есть никто в локальной сети не сможет подсмотреть за нами и украсть имена пользователей и пароли. Достоинство туннелирования в том, что оно позволяет делать это с любым видом web-трафика, не только с оболочками. Во-вторых, это позволяет обойти фильтрацию содержимого

»

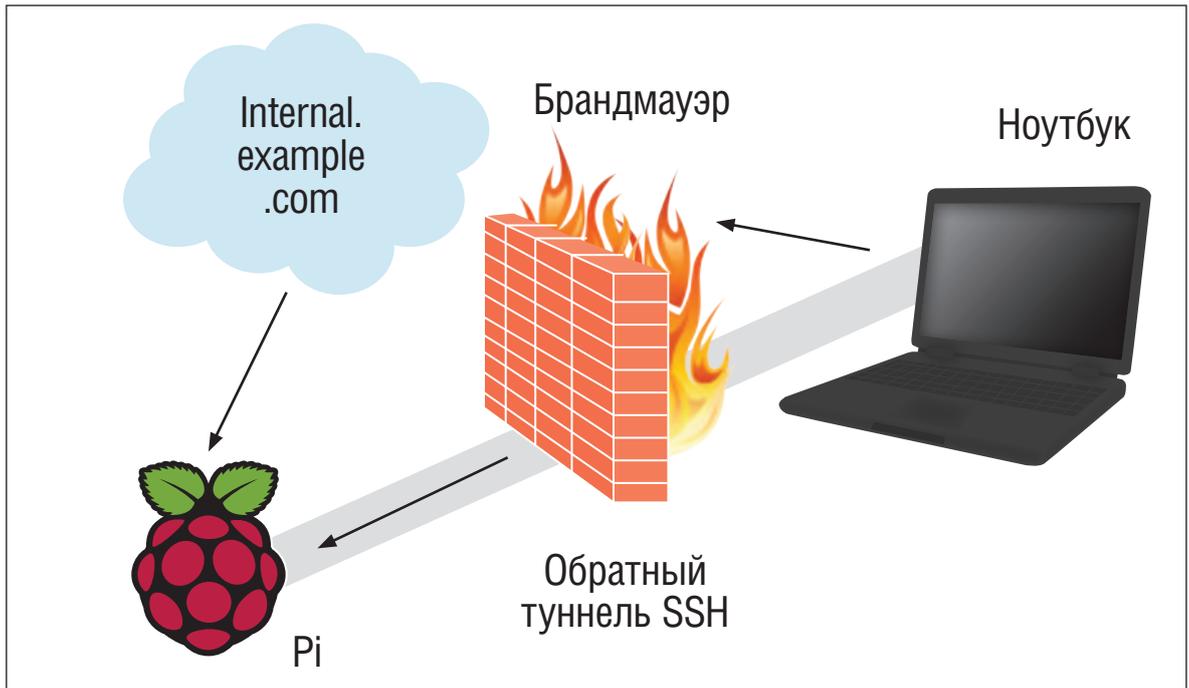
## Что такое SSH?

Если Raspberry Pi у вас не первый день, то вы знакомы с командной строкой, она же – оболочка. При первом запуске Pi появляется черно-белый экран, где нужно набрать свое имя пользователя и пароль. Затем вы предоставляетесь самим себе и можете выполнять всякие действия – вводить команды запуска графического интерфейса, редактирования текстовых файлов или запуска браузера вроде *ELinks*.

SSH – сокращение от “secure shell [защищенная оболочка]”. Через нее можно безопасно входить на удаленный компьютер – подключение будет защищенным, т.е. никто не сможет отследить пересылаемые данные, будь то имя пользователя и пароль, файлы или интернет-адреса и cookies. По сути это современная, защищенная версия старых утилит удаленного подключения к системе, типа *telnet* и *rlogin*.

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

» С помощью обратного туннеля можно проникнуть внутрь защищенных сетей. Только предупредите свой IT-отдел, а не то доставите ему беспокойство.



сайтов в локальной сети. Если маршрутизатор, за которым находится локальная сеть, фильтрует трафик на `example.com`, перенаправляя соединение через неотфильтрованный порт на неотфильтрованный компьютер, в локальной сети такие ограничения можно обойти. Для этого нужно убедиться, что SSH установлен и запущен в виде сервиса на `laptop` и `pi`. На `pi` для этого достаточно открыть утилиту `raspi-config` и выбрать опцию SSH (если у вас последняя версия Raspbian).

1024 можно перенаправлять без прав `root`. Весь трафик, который мы направим на этот порт, будет перенаправлен. Средний фрагмент – компьютер, куда мы хотим перебросить трафик, а последнее число – порт на этом компьютере, к которому мы хотим подключиться. Компьютер и порт могут быть любыми – если у `pi` есть к ним доступ, все будет работать. Так как мы хотели открыть сайт `example.com`, то указали порт 80 – по умолчанию он используется для `web`-трафика. В конце указываются имя пользователя и компьютер, через которые мы хотим пропустить туннель. Здесь это компьютер `pi` и пользователь `pi-user`.

Эта часть работает как в стандартной команде `ssh`, и здесь можно пользоваться всеми обычными приемами: например, указать другие порты, использовать IP-адреса вместо имен компьютеров и т.д. (помните, что компьютер должен быть доступен для `laptop`, т.е. у него должно быть доменное имя или публично доступный IP-адрес; советы, как добиться этого при домашнем подключении, приведены во врезке «Динамический DNS»).

Параметр `-f` велит `ssh` перейти в фоновый режим и не занимать терминал в `pi`.

Теперь, чтобы открыть сайт `example.com` на `laptop`, нужно лишь открыть в браузере адрес `http://localhost:1080`, и все должно получиться. Двоеточие в конце адреса означает, что нужно использовать указанный порт вместо порта по умолчанию 80. “localhost” означает, что нужно обратиться к порту 1080 на том же компьютере – это «адрес вашего компьютера в Интернете».

Можно легко изменить настройки, чтобы пользоваться и другими сервисами, а не только `web`-трафиком. Например, если на `pi` есть собственный почтовый сервер (SMTP), можно изменить `example.com` на `localhost` и 80 на 25, и все будет готово. Затем, вместо того, чтобы вводить специальный адрес в браузере, нужно изменить некоторые настройки в почтовом клиенте.

## Обратные туннели

SSH умеет создавать и другие туннели. Кратко рассмотрим их. Если в предыдущем примере у нас было перенаправление локального порта (`-L`), следующий можно назвать перенаправлением удаленного порта.

## «Чтобы обойти брандмауэр снаружи, можно создать обратный туннель с pi на laptop.»

На ноутбуке настройка зависит от дистрибутива. Для настройки SSH в Fedora выполните команды:

```
su -c "yum -y install openssh"
su -c "systemctl enable sshd.service"
su -c "systemctl start sshd.service"
```

А в Debian или Ubuntu –

```
sudo apt-get install openssh-server openssh-client
sudo service ssh start
sudo inserv ssh
```

Если на ноутбуке стоит Windows, понадобится *Putty* или *Cygwin*, но обе эти программы выходят за рамки нашей статьи.

## Создание туннеля

Сделав это, выполните следующую команду для создания туннеля:

```
ssh -L 1080:example.com:80 pi-user@pi -f
```

Разберем ее, чтобы вы поняли, что происходит. Параметр `-L` сообщает SSH, что мы будем использовать его для перенаправления порта.

Следующий фрагмент строки разбит на три части двоеточиями. Первое число – порт на локальном компьютере (`laptop`), который мы хотим перенаправить. Все порты с номером больше

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Представим себе несколько иной сценарий. У нас опять же два компьютера – **laptop** и **pi**. Ноутбук находится дома, но на нем нужно открыть внутренний сайт [internal.example.com](http://internal.example.com) какой-то другой сети, который блокируется для внешних подключений брандмауэром. А **pi** подключен ко внутренней сети.

Чтобы обойти этот брандмауэр снаружи, можно создать обратный туннель с **pi** на **laptop**. Затем с помощью этого туннеля ноутбук может проникнуть внутрь защищенной брандмауэром сети через **pi** и обратиться к [internal.example.com](http://internal.example.com).

Для этого на **pi** выполняется команда

```
ssh -R 1080:internal.example.com:80 laptop-user@laptop -f
```

Параметр **-R** означает обратный [reverse] или удаленный [remote] туннель. В следующем фрагменте текста первый параметр задает не локальный порт на **pi**, с которого нужно перенаправлять подключения, а порт ноутбука для той же цели.

Все остальное работает как раньше. Чтобы извлечь максимум из этой уловки, туннель нужно создать заранее, так как у вас не будет доступа к **pi** с ноутбука, и поддерживать SSH-соединение активным. Для этого вместо обычного SSH можно установить и воспользоваться **autossh**, оберткой, поддерживающей соединение активным. Синтаксис команды **autossh** несколько иной:

```
autossh -M 20000 -f 1080:internal.example.com:80 laptopuser@laptop
```

Здесь обратите внимание на параметр **-M**. Указанный порт не должен использоваться другими программами, поэтому выберите номер побольше. Он используется **autossh** для отслеживания соединения и его восстановления, если оно пропадает.

Правда, этот полезный прием может поссорить вас с системными администраторами. Брандмауэры ставят все-таки не зря, и администраторы не любят, когда другие проделывают в них дырки!

Некоторые специалисты в области безопасности используют именно эту технологию для проверки корпоративных сетей на прочность. С помощью методов социальной инженерии они проносят в здания компаний маленькие коробочки (и Raspberry Pi идеально подходит для этого), находят незанятые сетевые разъемы и подключают их к ним. Коробочки настраиваются на автоматическое создание SSH-туннеля (например, с помощью скрипта `/etc/network/if-up.d`), через который можно будет проникнуть в сеть компании, украсть пароли и поработать с файлами!

## Динамические туннели

Вот последняя и, пожалуй, самая полезная из технологий туннелирования SSH. Можно не указывать конкретные порты или сервисы для обмена информацией, а создавать динамические

туннели, способные собирать и отправлять данные всем сервисам. Команда в этом случае гораздо проще:

```
ssh -D 1080 pi-user@pi
```

Параметр **-D** означает, что мы хотим настроить динамический туннель, номер – используемый локальный порт, а **pi-user@pi** – пользователь и компьютер, к которым будет проложен туннель.

По выполнении этой команды можно настроить любую программу, поддерживающую прокси SOCKS, на отправку трафика через только что созданный SSH-туннель.

Например, в *Firefox* нужно зайти в Preferences > Advanced > Network [Настройки > Дополнительные > Сеть] и нажать на кнопку Settings [Настройки] в разделе Connection [Подключение]. В появившемся окне выберите ручную [Manual] настройку прокси. Затем введите в строку SOCKS host [Узел SOCKS] адрес **127.0.0.1** и порт, указанный в команде **ssh**.

Нажмите OK, и все должно быть готово. Если все работает правильно, сайты должны открываться как обычно, но теперь весь трафик будет передаваться через туннель SSH на компьютер **pi**.

Чтобы убедиться, что все действительно работает, закройте SSH-соединение (нажав Ctrl+c) и попробуйте открыть страницу снова. Если использовался туннель, то теперь в *Firefox* страницы открываться не будут.

В других программах, подключающихся к Интернету – например, почтовых клиентах и программах для обмена сообщениями, будут аналогичные окна с настройками. Просто заполните параметры SOCKS, и все будет работать.

## «Последняя и, пожалуй, самая полезная из технологий туннелирования SSH.»

На этом все. Продемонстрированные технологии очень удобны для создания защищенного и неограниченного подключения во время поездок, но для их работы сервер должен быть постоянно включен. Электроэнергия дорожает, но с Raspberry Pi в качестве сервера расходы будут невелики.

Если вам нужно сделать сервисы в локальной сети доступными удаленным пользователям, вам могут быть интересны и VPN – виртуальные частные сети (Virtual Private Network). SSH проще в настройке, но он и менее гибок. Если вам интересно это направление, почитайте вводную статью «Применим VPN» в **LXF158**. 

## Динамический DNS

В большинстве методов, описанных в этой статье, используется компьютер с общедоступным IP-адресом или доменным именем. Большинство провайдеров предоставляют общедоступный IP-адрес, но он динамический, т.е. время от времени меняется. И если вы настроите туннель, а потом IP-адрес окажется другим, все перестанет работать.

Есть несколько вариантов решения проблемы. Самый простой – найти провайдера, который выделит вам статический IP-адрес. Некоторые выделяют его по вашей просьбе, другие просят за это дополнительную плату, третьи не делают этого ни при каких обстоятельствах.

Если сменить провайдера не получается, можно воспользоваться такими сервисами, как [www.no-ip.com](http://www.no-ip.com) и [www.dyn.com](http://www.dyn.com). С ними вы запускаете

программу, которая отслеживает ваш IP-адрес и при его изменении оповещает сторонний сервис. Сервисы предоставляют вам доменное имя вроде `myip.dyn.com`, которое всегда указывает на IP-адрес вашего компьютера, независимо от того, сколько раз он меняется.

Если Raspberry Pi находится за маршрутизатором, убедитесь, что порт 22 перенаправляется на него. Конкретные действия по настройке немного отличаются у разных маршрутизаторов, но обычно у них есть веб-интерфейс администратора, зайдя в который, можно перенаправлять порты на компьютеры по их MAC-адресам (аппаратным адресам). Также можно установить для Pi статический IP-адрес и использовать его как место перенаправления порта.



**no-ip.com** создает динамическую запись DNS для доступа к Pi, даже если его IP-адрес изменился.

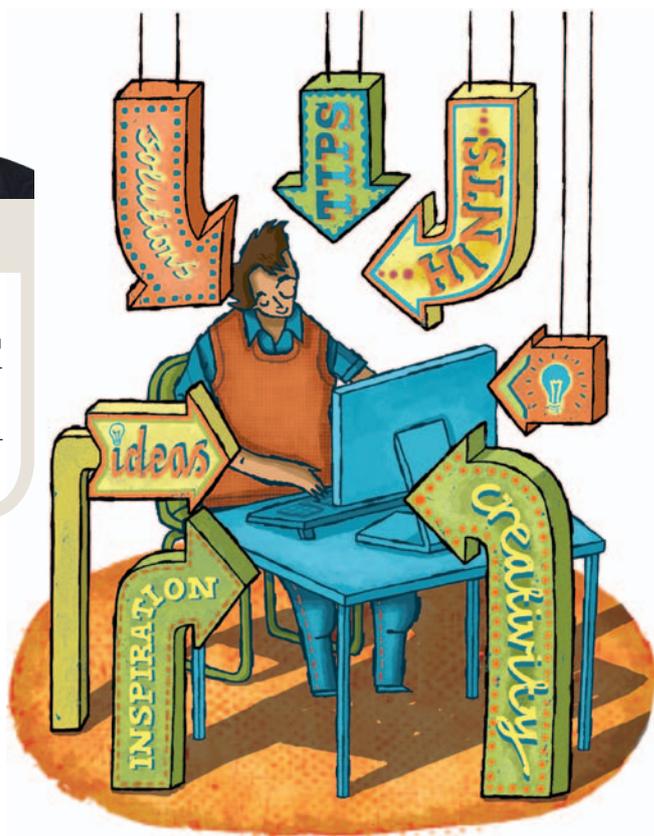
# Пароли: ЕСТЬ

Вы когда-нибудь думали о стратегии выбора идентификатора пользователя и пароля? Джеймс Литтон подбросит вам пиццы для размышлений.



Наш эксперт

Джеймс Литтон занимал руководящие должности во многих организациях. Ныне он совладелец и исполнительный директор Identity Automation LP.



С момента появления первых многопользовательских систем в аутентификации пользователей мало что изменилось. У нас все еще есть идентификатор пользователя (ID) и пароль – и если они верны, мы предоставляем доступ (т.е. авторизацию), и пользователь берется за дело.

Кажется, каждую неделю нам сообщают о дешевых и общедоступных вычислительных ресурсах современных процессоров и графических сопроцессоров, способных выполнять столько операций в секунду, что всего пару лет назад это казалось невыполнимым. Когда такие ресурсы направляются на взлом паролей, это часто позволяет «плохим парням» добраться до ваших данных.

## Защитимся

Предположим, вы администрируете сервер, доступный через Интернет, и на нем запущено несколько сервисов. Что, спросите вы, можно сделать для снижения моих шансов стать жертвой атаки? Рассмотрим же некоторые простые стратегии и готовые утилиты, которые помогут уменьшить ваши риски! Первое и главнейшее – подумайте об основах. Имена пользователей и пароли – ключи для входа в систему; вот с них и начнем. Так как аутентификация в системе основана на идентификаторах пользователей и паролях, очевидно, что даже имя пользователя – важная часть схемы безопасности организации. В большинстве организаций иденти-

фикатор пользователя определить легко – например, это может быть первая буква и фамилия пользователя (vivapov) или имя и фамилия (vasyavanov). Если мы хотим создать препятствия для «плохих парней», то такой подход – не лучшая практика. Технологии социальной инженерии, публикации и другие источники и методы позволяют легко узнать имена лиц, которых стоит атаковать. Лучше иметь такие идентификаторы пользователей, по которым пользователей нельзя определить.

В некоторых организациях за идентификаторы пользователей берутся табельные номера сотрудников или студентов. Это гораздо лучше методов, основанных на персональной информации, но прежде чем применять такой подход, хорошенько подумайте. Ответьте на следующие вопросы:

- » Напечатаны ли эти номера на бейджиках?
- » Используются ли они для другой идентификации?
- » Идут ли они подряд?

Еще один метод – генерировать идентификаторы для всех пользователей случайным образом. Тогда два лица, поступившие на работу в одном месте в один и тот же день, получат совершенно разные и неузнаваемые идентификаторы.

Джо Смит 947G25  
Ким Джонс 21003N

При использовании этого метода единственное, что связывает идентификаторы с лицами – сам процесс назначения идентификаторов. Это дает нам фору перед «плохими парнями», раз мы не даем им способа получить один из двух компонентов, необходимых для аутентификации.

Есть несколько способов генерации случайных идентификаторов, но я люблю простоту, так что мы здесь воспользуемся скриптом *Bash*.

Для начала создадим файл, где будут записаны команды, и сделаем его исполняемым:

```
cd ~  
touch genid.sh  
chmod +x genid.sh
```

Теперь откройте **genid.sh** в любимом текстовом редакторе и добавьте следующие команды:

```
#!/bin/bash  
I=$1  
[ "$I" == "" ] && I=6  
tr -dc A-Z0-9 < /dev/urandom | head -c $I | xargs
```

Для проверки скрипта просто наберите **./genid.sh** в командной строке, и вы получите случайную строку из шести символов, скажем, **P3NVUE**.

При каждом запуске скрипт формирует новую случайную строку. Длину выходной строки можно менять, передав ее как параметр. Например, команда **./genid.sh 10** выдаст случайную строку из десяти символов.

Первая строка кода говорит оболочке, что последующие строки должны интерпретироваться *Bash*. Во второй строке первый параметр, переданный скрипту, сохраняется в переменной **I** для дальнейшего употребления. В третьей строке **I** устанавливает-

# Стратегия лучше

ся в 6, если параметра передано не было. Наконец, в четвертой строке с помощью генератора случайных чисел Linux создается случайный поток символов, а затем команда транслитерации заменяет эти символы на буквы верхнего регистра и цифры. Затем команда **head** выделяет количество символов, заданное переменной **l**, и, наконец, **xargs** дает нам символ конца строки.

В третьей строке можно задать длину выходной строки по умолчанию, присвоив значение параметру **l**. По моему опыту, идентификатор из шести символов очень легко запомнить.

## Пароли

За последние 15–20 лет написана тьма статей о паролях, с советами оставлять их как очень простыми, так и очень сложными. Большинству из нас знакомы сложные требования организаций к паролям, которые пугают пользователей и заставляют их записывать пароли на бумаге, чтобы не забыть.

Основной целью политики выбора паролем должно стать сохранение их достаточной сложности, чтобы создать реальный барьер злоумышленникам и в то же время не отпугнуть пользователей.

В качестве аналогии попробуем представить себе входную дверь. Она может быть в одном из двух состояний: заперта или незаперта. Если совсем убрать пароли – верите вы или нет, желающие воспользоваться этой возможностью найдутся. Кто-то сможет получить доступ к учетной записи, просто введя идентификатор пользователя. Это аналогично незапертой двери. Продолжая аналогию, все мы знаем, что замки бывают разные – в дверной ручке с кнопкой, засовы с ключом или замки, применяемые в банковских ячейках (ну, это чересчур, но вы поняли идею).

**«В качестве аналогии попробуем представить входную дверь.»**

## Энтропия — у паролей?!

Энтропия пароля – это оценка его близости к случайному и, следовательно, трудности его угадывания. Энтропия обычно выражается в битах. Например, известный пароль содержит ноль бит энтропии. Пароль, который можно угадать с первой попытки в 50% случаев, имеет один бит энтропии.

Следовательно, пароль с *n* битами энтропии угадать так же сложно, как *n*-битное случайное число. То есть, пароль с *n* битами энтропии можно угадать за 2<sup>*n*</sup> попыток. Более подробная информация приведена в руководстве по электронной аутентификации Национального института стандартов США на сайте [1.usa.gov/q3y5u](http://1.usa.gov/q3y5u).

```
File Edit View Terminal Help
root@littton01:~# faillog -a
Login      Failures Maximum Latest      On
-----
root       0          3    12/31/69 18:00:00 -0600 [3600s lock]
daemon    0          3    12/31/69 18:00:00 -0600 [3600s lock]
bin       0          3    12/31/69 18:00:00 -0600 [3600s lock]
sys       0          3    12/31/69 18:00:00 -0600 [3600s lock]
sync      0          3    12/31/69 18:00:00 -0600 [3600s lock]
games     0          3    12/31/69 18:00:00 -0600 [3600s lock]
man       0          3    12/31/69 18:00:00 -0600 [3600s lock]
lp        0          3    12/31/69 18:00:00 -0600 [3600s lock]
mail      0          3    12/31/69 18:00:00 -0600 [3600s lock]
news      0          3    12/31/69 18:00:00 -0600 [3600s lock]
uuuicp    0          3    12/31/69 18:00:00 -0600 [3600s lock]
proxy     0          3    12/31/69 18:00:00 -0600 [3600s lock]
www-data  0          3    12/31/69 18:00:00 -0600 [3600s lock]
backup    0          3    12/31/69 18:00:00 -0600 [3600s lock]
list      0          3    12/31/69 18:00:00 -0600 [3600s lock]
irc       0          3    12/31/69 18:00:00 -0600 [3600s lock]
gnats     0          3    12/31/69 18:00:00 -0600 [3600s lock]
nobody    0          3    12/31/69 18:00:00 -0600 [3600s lock]
libuid    0          3    12/31/69 18:00:00 -0600 [3600s lock]
syslog    0          3    12/31/69 18:00:00 -0600 [3600s lock]
mysql     0          3    12/31/69 18:00:00 -0600 [3600s lock]
messagebus 0          3    12/31/69 18:00:00 -0600 [3600s lock]
landscape 0          3    12/31/69 18:00:00 -0600 [3600s lock]
sshd      0          3    12/31/69 18:00:00 -0600 [3600s lock]
jlitton   3          3    07/07/12 17:53:26 -0500 [1060s left]
root@littton01:~#
```

Требование задать пароль, не оговоренное правилами, даст нам слишком простые пароли, которые легко подобрать. Примеры – имя человека, его ребенка или котенка и классика вроде password или abc123. Это аналогично простейшему замку в дверной ручке: дверь заперта, но не особо надежно. Другая крайность – попытка достичь уровня защиты банковской ячейки, задавая сложные правила. Но это приводит к чрезмерному усложнению паролем – и из-за невозможности их запоминания пользователи примутся их записывать, создавая дополнительный риск. Нужна золотая середина.

Я бы сказал, что наша цель – сделать пароль достаточно сложным для взлома и не слишком сложным для запоминания пользователем, и одновременно применить другие средства для предотвращения атак методом перебора и других вредоносных действий.

## Сила пароля: факт или фикция?

Измерение надежности пароля по количеству бит его энтропии, согласно описанию Национального института стандартов и технологий, говорит нам, что пароль из трех или четырех несвязанных слов обладает той же степенью надежности, что и более короткий пароль из необычных символов.

Возьмем слова «яблоко», «айфон», «нога» и «поезд» и объединим их – мы получим «яблокоайфонногапоезд». Такой пароль неплохо поддается запоминанию, а «плохие парни» вряд ли легко его разгадают. Сравним это со сложными политиками паролей, »

» Настройка pam\_tally.

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

заставляющими пользователей придумывать экзотику вроде `Ff%#ht!4-2`. Этот пароль не только не упомянуть, но и с точки зрения энтропии он менее надежен, чем наш.

Чтобы представить себе это более наглядно, наш второй пароль, который так сложен и вроде бы надежен, имеет 27 бит энтропии, и теоретически его можно взломать всего за 37 часов при скорости перебора 1000 вариантов в секунду. Наш первый пароль, который гораздо проще запомнить и который, на первый взгляд, менее надежен, имеет 34.5 бита энтропии, и на его взлом с той же скоростью перебора ушло бы около 281 дня. Добавим к нему символ в другом регистре и спецсимвол, получив что-то вроде «яблокоайфонНога#поезд», и мы увеличим энтропию до 42 бит. На взлом такого пароля потребуется уже 139 лет.

Итак, сложнее не всегда значит лучше. Пароля длиной 8–15 символов верхнего и нижнего регистра более чем достаточно, чтобы запереть входную дверь системы на надежный засов.

Создадим еще один скрипт *Bash* – пусть предлагает нам слова для паролей. Сначала создадим файл скрипта и сделаем его исполняемым:

```
cd~
touch genpwd.sh
chmod +x genpwd.sh
```

Теперь нам нужен файл словаря, откуда будут браться слова. Скорее всего, такой файл в вашей системе уже есть. Это файл **words** в каталоге `/usr/dict/` или `/usr/share/dict/`. Если его нет, список доступных словарей в Ubuntu можно посмотреть, набрав `apt-get install wordlist` в командной строке. Например, большой словарь американского английского можно установить командой `sudo apt-get install wamerican-large`.

Для примера я воспользуюсь файлом с 1000 самых ходовых английских слов.

```
cd ~
wget www.rupert.id.au/resources/1-1000.txt
```

Создайте файл **genpwd.sh** в своем любимом текстовом редакторе и введите в него следующие строки:

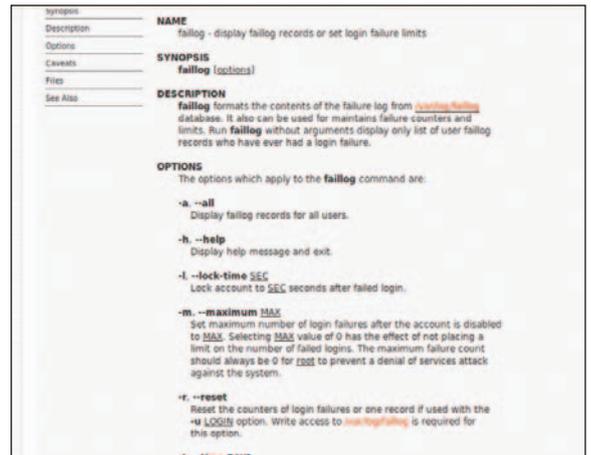
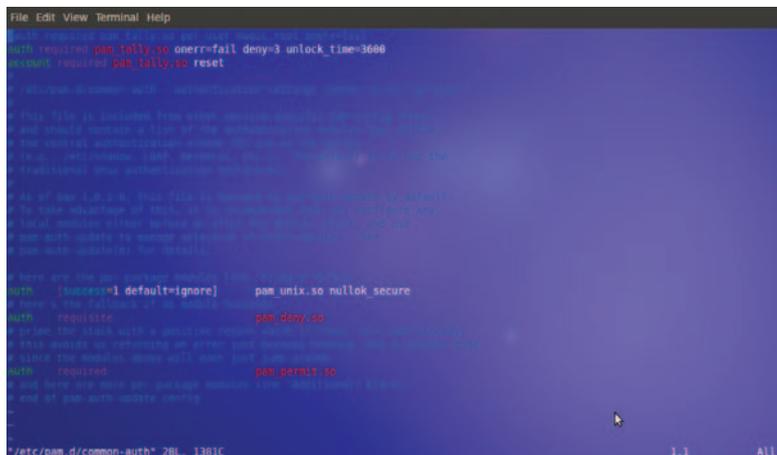
```
#!/bin/bash
shuf ~/1-1000.txt | head -n4
```

Для проверки скрипта просто наберите `./genpwd.sh` в командной строке, и вы получите четыре случайных слова, например:

- » fell
- » wife
- » substance
- » excite

Неплохое начало! Теперь соединим слова и получим легкий для запоминания пароль.

» Проверка неудачных входов в систему с помощью faillog.



» Добавочную информацию по работе с faillog можно найти на ман-странице.

Как видите, скрипт очень прост. Как и в первом скрипте, первая строка означает, что следующие строки должны обрабатываться оболочкой *Bash*. Во второй строке указана команда **shuf**, которая перемешивает строки в файле словаря. В команде **head** мы указываем количество слов из файла, которые нужно получить.

## Идем дальше

Возможно, вы захотите пойти дальше и ограничить длину слова, чтобы в нем было от трех до пяти символов. Для этого измените вторую строку:

```
shuf ~/1-1000.txt | grep "^[^]{3,5}$" | head -n4
```

Здесь мы дополнили команду **grep** строкой **regex**, которая выбирает только слова заданной длины.

Я бы предпочел не ограничивать длину слова, но хочу, чтобы слова выводились в одной строке вместе и каждое слово начиналось с заглавной буквы. Для этого измените вторую строку:

```
shuf ~/1-1000.txt | head -n4 | sed 's/[^ ]+/\U&/g' | tr -d '\n' | xargs
```

Эта команда выдает строки вроде **SkyLoveStartBy** (33-битный легкий для запоминания пароль).

В этом примере мы берем код первого скрипта, делаем первую букву каждого слова заглавной командой **sed** и удаляем переносы строк командой **tr**, поэтому все слова теперь будут в одной строке. Затем с помощью **xargs** мы приписываем к этой строке символ конца строки.

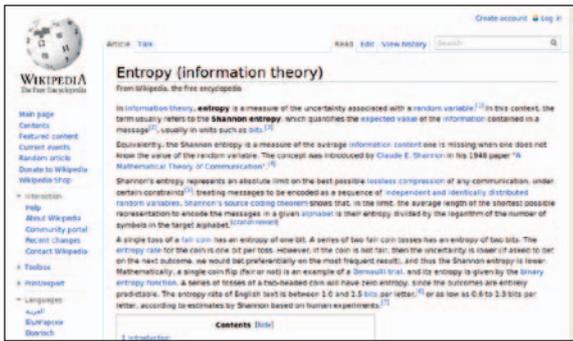
## Мониторинг неудачных попыток

Лучший способ ограничить риск ущерба от атак методом перебора – отслеживать неудачные попытки входа в систему и принимать меры при их обнаружении. Эти меры принимаются во многих системах и могут включать различные действия от временной или постоянной блокировки атакуемой учетной записи до блокирования доступа для источника атак.

## Длиннее = лучше

Разумеется, чем длиннее пароль, тем сложнее его подобрать. Пароль длиной 10 символов из букв и цифр имеет почти квинтиллион (839 квадриллионов) возможных вариантов. Задайте длину 15 символов – и число вариантов приблизится к одному октиллиону (768 септиллионов). Это ну очень много!

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.



➤ Измерение энтропии для определения, является ли “grasshatfall” более случайным, чем “5@PU”.

Простейший способ отслеживать неудачные попытки входа в систему – воспользоваться модулем `pam_tally` Linux-PAM (Plugable Authentication Modules – Подключаемые модули аутентификации). В моей Ubuntu 11.10 Server я подключил `pam_tally`, открыв файл `/etc/pam.d/common-auth` и добавив в его начало две следующие строки:

```
auth required pam_tally.so onerr=fail deny=3 unlock_time=3600
account required pam_tally.so reset
```

С помощью команды `faillog` можно управлять счетчиками, просматривать информацию о неудачных попытках входа в систему с момента последнего входа в систему, о блокировках и о продолжительности текущих блокировок. Чтобы `faillog` отображала время блокировки, настройте ее в соответствии с параметром `unlock_time`, заданным в файле `/etc/pam.d/common-auth`. Для этого скомандуйте `faillog -i 3600`

Теперь, если набрать `faillog` в командной строке, для всех заблокированных учетных записей выведется время, оставшееся до разблокирования модулем `pam_tally`, в секундах. Этот метод мониторинга распространяется на вход в систему через консоль, а также по `ssh`.

Еще одна утилита, которая навсегда блокирует попытки входа в систему с заданного хоста, называется `fail2ban`; это фреймворк для предотвращения вторжений, написанный на Python.

Описание работы с `fail2ban` выходит за рамки этой статьи, но соответствующая информация имеется на сайте [help.ubuntu.com/community/Fail2ban](http://help.ubuntu.com/community/Fail2ban).

Лучший способ достичь максимальной уверенности в том, что человек действительно тот, за кого себя выдает – выполнить проверку личности в какой-либо форме. Если у вас когда-либо просили предъявить водительские права, студенческий билет, ввести номер социального страхования или еще какую-то персональную информацию для завершения транзакции, то вы уже проходили проверку личности. Проверка личности – просто просьба по отношению к кому-либо идентифицировать себя, доказав, что он – и вправду он. В мире аутентификации это часто называется двухфакторной аутентификацией.

## Пассивные системы

Для проверки лучше всего пользоваться пассивной системой, которая не требует от пользователя лишних действий или применения дополнительных устройств. Одна из таких систем – динамика нажатия клавиш – использует алгоритмический подход для оценки вероятности того, что вы – тот, кто вы есть. В ней измеряется ритм нажатия клавиш при вводе пароля. Таким образом составляется уникальный биометрический шаблон для каждого пользователя, и эти данные применяются для сравнения при следующих попытках аутентификации.

На практике эти системы работают в фоновом режиме: проверяют, как вводятся пароли, и разрешают или запрещают вход по результату сравнения. Этот подход – прекрасный способ отражения атак компьютера и человека.

Еще одна форма проверки, набравшая некоторую популярность в последние годы – биометрическая: сканируется радужная оболочка глаза или отпечаток пальца и результат сравнивается с хранящимся в базе данных. Эти системы больше вторгаются в частную жизнь, но их преимущество в том, что для них не требуется дополнительных устройств, таких как USB-брелок, карта или телефон.

Самые популярные и, пожалуй, самые практичные системы проверки личности – генераторы одноразовых паролей, которые требуют от пользователя указывать уникальную строку символов при каждом входе в систему. Но тогда пользователи обычно должны носить с собой дополнительные устройства. В более новых системах используются приложения на мобильных устройствах (телефонах) и подтверждение при посредстве SMS, для максимальной надежности.

Я не рассказывал об этом в нашей статье, но в качестве двухфакторной аутентификации в Linux вы можете попробовать программу Google Authenticator. **LXF**

## «На взлом нашего первого пароля ушел бы 281 день.»

| Name                             | Short scale (U.S. and modern British) | Long scale (continental Europe, older British) | Authorities         |                    |                     |                       |                     |                      |                   |                   |
|----------------------------------|---------------------------------------|--|---------------------|--------------------|---------------------|-----------------------|---------------------|----------------------|-------------------|-------------------|
|                                  |                                       |  | AHD4 <sup>[1]</sup> | COD <sup>[2]</sup> | OED2 <sup>[3]</sup> | OEDnew <sup>[4]</sup> | RHD2 <sup>[5]</sup> | SOED3 <sup>[6]</sup> | W3 <sup>[7]</sup> | UM <sup>[8]</sup> |
| Million                          | 10 <sup>6</sup>                       | 10 <sup>6</sup>                                | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Milliard                         |                                       | 10 <sup>9</sup>                                | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Billion                          | 10 <sup>9</sup>                       | 10 <sup>12</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Trillion                         | 10 <sup>12</sup>                      | 10 <sup>15</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Quadrillion                      | 10 <sup>15</sup>                      | 10 <sup>18</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Quintillion                      | 10 <sup>18</sup>                      | 10 <sup>21</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Sextillion                       | 10 <sup>21</sup>                      | 10 <sup>24</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Septillion                       | 10 <sup>24</sup>                      | 10 <sup>27</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Octillion                        | 10 <sup>27</sup>                      | 10 <sup>30</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Nonillion                        | 10 <sup>30</sup>                      | 10 <sup>33</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Decillion                        | 10 <sup>33</sup>                      | 10 <sup>36</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Undecillion                      | 10 <sup>36</sup>                      | 10 <sup>39</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Duodecillion                     | 10 <sup>39</sup>                      | 10 <sup>42</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Tredecillion                     | 10 <sup>42</sup>                      | 10 <sup>45</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Quattuordecillion                | 10 <sup>45</sup>                      | 10 <sup>48</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Quindecillion (Quinquadecillion) | 10 <sup>48</sup>                      | 10 <sup>51</sup>                               | ✓                   | ✓                  | ✓                   | ✓                     | ✓                   | ✓                    | ✓                 | ✓                 |
| Sextodecillion                   |                                       |  |                     |                    |                     |                       |                     |                      |                   |                   |

➤ Таблица потребления порций алкоголя Грэмом и Эндрю на конференции OSCP.

## Полная безопасность

Безопасность – это не только имя пользователя и пароль. Если ваш сервер открыт для доступа через Интернет, защитите его надежным брандмауэром, дающим доступ лишь к минимально необходимому

набору сервисов. Также имеет смысл запустить брандмауэр на других серверах и отключить ненужные сервисы. Наконец, если вы работаете системным администратором и вам необходимо регулярно

подключаться по `ssh` ко внутренним серверам, подумайте о применении аутентификации на базе сертификатов, взамен традиционной парольной аутентификации.



# Облако файлов:

Нейл Ботвик рассматривает опции облачного хранения – на тот случай, если катастрофа постигнет вашу домашнюю систему.



## Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не скажет, где находится центральный сервер.

Все согласны, что резервные копии – это хорошо, и некоторые даже их делают. Большинство делает их на внешний жесткий диск, что защитит вас при отказе основного жесткого диска, но мало поможет в более катастрофической ситуации – пожара, наводнения, кражи со взломом, а то и просто скачка напряжения или его кратковременного провала в момент подключения внешнего жесткого диска. Локальные резервные копии удобны и делаются быстро, но ненадежны. Внешние резервные копии имеют свои ограничения: скорость, стоимость, безопасность хранения ваших данных на чужом компьютере. Мы рассмотрим, как обеспечить сохранность ваших данных удаленным (или облачным, кому как нравится) хранением. Так как локальные резервные копии более удобны, если они под рукой, мы выберем подход, когда эти копии любым привычным вам способом синхронизируются с облачным сервером, вместо того, чтобы делать дубликаты, настраивая отдельный сервер резервного копирования.

Есть множество сервисов облачного хранения; и некоторые из них предоставляют «жилплощадь» бесплатно – с платными обновлениями за ее расширение. Это хорошие условия, если вам нужно меньше места, чем предоставляемое бесплатно, или вы точно знаете, сколько места вам понадобится, прежде чем платить за него. Здесь мы рассмотрим сервис, который в большей степени использует метод оплаты по факту использования – сервис *S3* от Amazon. Он не предоставляет клиента рабочего стола, все делается через web-интерфейс; но здесь есть API, так что желающие могут написать клиента. Для *Amazon S3* имеется кроссплатформенный GUI-клиент *DragonDisk*, доступный на <http://www.dragondisk.com> или через менеджер пакетов вашего дистрибутива. Однако резервные копии должны делаться автоматически в фоне и без вмешательства пользователя – ибо тот не хочет откладывать подобное вмешательство, вспоминая о нем исключительно тогда, когда резервная копия нужна. Выбор здесь имеется, но мы остановились на *s3cmd*, программе командной строки, объединяющей функции *ftp* (программы, а не протокола) и *rsync*.

## Откройте учетную запись

Первым делом нужно настроить учетную запись в *Amazon S3*. *S3* не предоставляет бесплатных услуг в том виде, как, например, *Dropbox*, т.е. не выдает вам 2 ГБ навеки, но в нем есть бесплатный пробный период. Он предоставляет вам 5 ГБ хранилища

и 15 ГБ в месяц бесплатной передачи, сроком на год или до тех пор, пока вы не превысите бесплатного лимита. Однако цены *S3* весьма разумны и основаны только на реальном использовании. Вы платите за используемое место и за ширину полосы, используемой для передачи, хотя последняя цена минимальна, если вы всего лишь загружаете резервные копии через равные промежутки времени. Чтобы завести счет, вам придется указать информацию по вашей карте, но если вы уже делали покупки на Amazon, это, скорее всего, уже сделано. Для создания счета перейдите на <http://aws.amazon.com/s3>. Когда счет будет создан, вы получите два ключа: ключ доступа и секретный ключ; по сути это имя пользователя и пароль. Справочная man-страница *s3cmd* говорит, что их надо разместить в `~/.s3cmd`, но файл, по крайней мере для релиза 1.1., будет `~/.s3cfg`. Впрочем, это особой роли не играет, поскольку *s3cmd* создаст файл за вас, когда вы запустите `s3cmd --configure`

Ответьте на вопросы, вырежьте и вставьте ключи, чтобы обеспечить их правильность, и *s3cmd* готов к работе. Запуск

```
s3cmd ls
```

не должен ничего отображать, если вы пока ничего не загрузили на свою учетную запись *S3*. Если он выдаст вам ошибку, проверьте настройку. *Amazon S3* работает с корзинами [buckets] и объектами [objects]. Корзина – это контейнер. Их может быть до 100, но их вложение не допускается: *S3* не поддерживает подпапок, но это не проблема, как мы увидим далее. Корзина создается из консоли web-управления, или с помощью *s3cmd*:

```
s3cmd mb s3://bucket-name
```

Названия корзин должны быть уникальными во всем *S3* – попытавшись создать корзину с уже использованным именем, вы получите ошибку. Можно избежать подобных коллизий, если предварять названия корзин своим доменным именем, если оно у вас есть; вряд ли кто-то еще возьмет, например, такое имя

```
s3cmd mb s3://example.com-backups
```

Мы отметили, что *S3* не поддерживает вложенных корзин, и в них могут быть включены только объекты (то есть файлы); так как же нам создавать резервные копии в папках или упорядочивать многочисленные копии, не создавая при этом множества корзин? Поскольку *S3* не использует папок, / является законным символом в имени файла. Итак, когда вы сохраняете файл как *somedir/somefile*, это один файл в корне вашей корзины, но *s3cmd* будет интерпретировать его, как файл в папке. Чтобы загрузить такой файл, введите:

```
s3cmd put somedir/somefile s3://example.com-backup/somedir/somefile
```

Полезным побочным эффектом того, что все это – один файл, является то, что вам не нужно создавать «папок» на сервере до того, как вы начнете в них писать. Вы также можете скопировать сразу несколько файлов в удаленную псевдопапку:

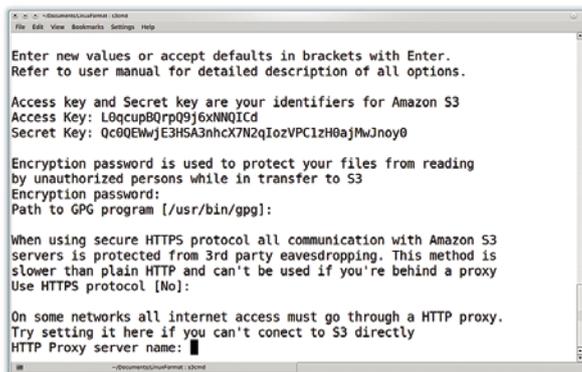
```
s3cmd *.jpg s3://example.com-backup/photos/
```

Копирование файла за файлом надоедает очень быстро; лучше мы синхронизируем всю папку резервных копий с *S3*:

```
s3cmd sync /mnt/backups/ s3://bucket-name/backups/
```

Если вам кажется, что это похоже на синтаксис для *rsync*, то вы правы, вплоть до значения слэшей в именах пути. Если

➤ Запустите `s3cmd --configure` для настройки. На данный момент все, что вам нужно изменить в настройке по умолчанию – это ключи.



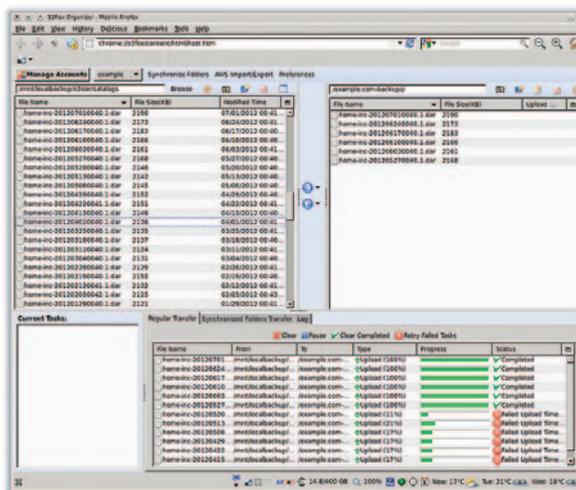
# Резервные копии

вы работали с *rsync*, то в опциях синхронизации *s3cmd* вы будете как дома. Например, можно использовать `--exclude` для указания папок или файлов, исключаемых из синхронизации, или `--exclude-from` для чтения этой информации из файла. Последняя опция полезна при создании резервных копий множественных путей в *S3*, поскольку можно указывать исключения отдельно для каждого пути и использовать тот же самый синтаксис, когда вызываете *s3cmd*. Одной значительной разницей между *rsync* и *s3cmd* – то, что *rsync* использует `--delete`, а *s3cmd* использует `--delete-removed`. Другое – то, что *s3cmd* автоматически перенаправляет в папки, если его вызвать с помощью *sync*. Объяснение этих опций можно найти на map-странице.

## Сравнение и контраст

Еще одно различие в том, как две команды изучают существующие файлы, обнаруживая в них различия: загрузка всех существующих файлов подряд – это явно не вариант, поэтому для их сравнения *s3cmd* использует два метода. Во-первых, сравниваются размеры файлов; а если они одинаковы, сравниваются контрольные хэши MD5 локальных и удаленных файлов. Вычисление удаленного хэша делается на сервере Amazon, и ваша полоса пропускания не нагружается. Однако при этом происходит небольшая задержка – еще одна транзакция с сервером (*S3* начисляет плату за каждые 1000 или 10000 транзакций), так что вы можете отключить эту операцию с помощью `--no-check-md5`. Для определения идентичности файлов часто хватает проверки их размера.

При использовании любого облачного хранения нужно учитывать безопасность ваших данных. Если вы создаете резервную копию своей домашней папки, там наверняка будет конфиденциальная информация. Сервиса, который шифрует ваши данные, недостаточно, потому что для этого ему придется указать ключи. Если вам нужна программа резервного копирования с собственным шифрованием, то лично мне показалась превосходной *dar* (<http://dar.linux.free.fr>). В ином случае вам придется рассмотреть возможность шифрования данных перед их загрузкой. Это можно сделать с помощью чего-то вроде *GPG*, но тогда придется хранить



► Расширение для *Firefox*, *S3fox*, дает хороший контроль над вашим хранением данных *S3*; в последнее время я использую *Firefox* исключительно для этого.

два экземпляра ваших резервных копий, если только вы не избавились от незашифрованной локальной копии после загрузки на сервер хранения. А тогда вам придется дешифровать все перед восстановлением – так что все равно понадобится лишнее место. Одно из решений – виртуальная файловая система *ecryptfs*: она хранит зашифрованные данные на вашем диске, но делает дешифрованную копию без необходимости удваивать дисковое пространство или заниматься дешифрованием вручную. Скажем, у вас есть две пустых папки в `/mnt/backup`, под названием **plain** и **secret**. Настройте файловую систему *ecryptfs* с помощью:

```
sudo mount -t encryptfs /mnt/backup/secret /mnt/backup/plain
```

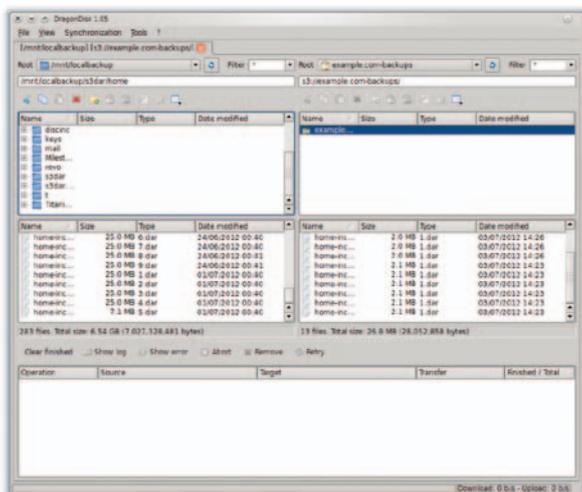
Задайте кодовую фразу для безопасности и примите настройки по умолчанию для прочих вопросов. Затем скопируйте пару файлов в **plain** и осмотрите содержание обеих папок. Вы увидите совпадающие файлы (шифровать имена файлов тоже можно), но файлы в **secret** будут нечитаемыми. Теперь размонтируйте:

```
sudo umount /mnt/backup/plain
```

Папка **plain** будет пуста, а **secret** будет по-прежнему содержать зашифрованные файлы. Повторите команду **mount**, чтобы вернуть назад содержимое. Осталось сделать резервную копию папки **plain** и синхронизировать папку **secret** с облаком:

```
s3cmd sync /mnt/backup/secret/ s3://bucket-name/backups/
```

Чтобы восстановить весь набор своих резервных копий, просто поменяйте в этой команде местами локальный путь и путь корзины. Резервные копии бывают весьма объемны, а в *S3* нет опции **resume** при загрузке данных, и хорошо бы ваша программа резервного копирования делила резервные копии на более мелкие части; тогда, если соединение разорвется, процесс придется повторить только для части копий. Стоит также повторить команду **sync** по завершении передачи данных; предпочтительнее сделать это с помощью опции `--no-check-md5`, чтобы удостовериться, что все было загружено без повреждений. Второй этап *s3cmd sync* должен завершиться без передачи файлов. *S3cmd* отлично подходит для передачи данных партиями из командной строки или скрипта резервного копирования, но иногда файлы надо просмотреть. Помимо уже упоминавшегося *DragonDisk* и *Amazon Web Console*, есть также модули расширения браузеров для *Firefox* и *Chrome/Chromium*; особо полезен *S3fox*. LXF



► *DragonDisk* – кросс-платформенная программа Java для работы с сервисом хранения *Amazon S3*.

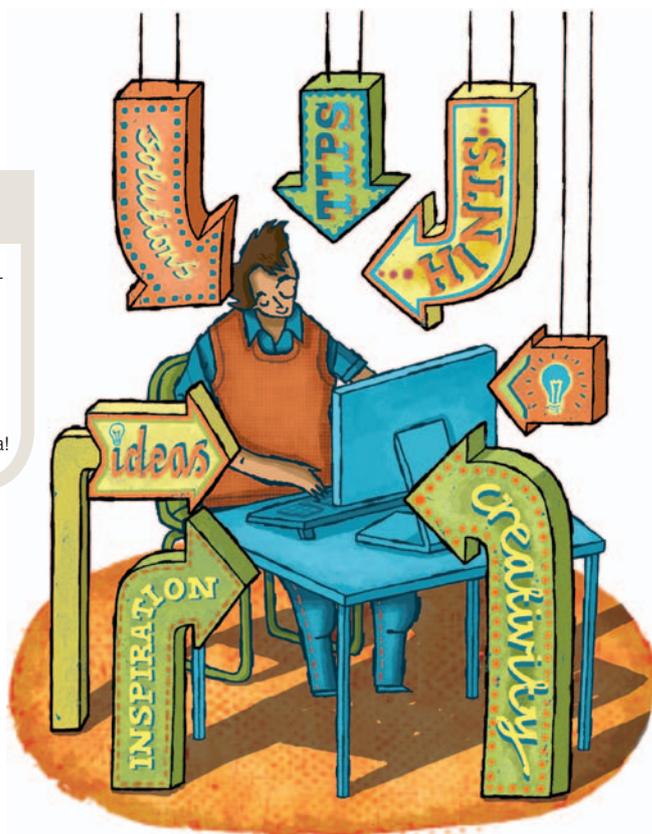
# Документация:

Улучшите жизнь пользователям своих программ или чужих программ, у которых нет хороших руководств. **Майк Сондерс** показывает, как это делается.



**Наш эксперт**

**Майк Сондерс** потратил больше времени на создание документации для своей ОС (<http://mikeos.berlios.de>), чем на написание кода!



**П**исать код здорово – можно оригинальничать, пробовать новое, выводить все из строя и похлопать себя по спине, когда что-то получится. А вот писать документацию не столь увлекательно, особенно когда вместо этого вы бы лучше реализовывали всякие функции. Это одна из причин, по которой документация в Linux (не считая популярных настольных приложений) часто неполна. Проще говоря, мало кому охота ее писать.



Однако хорошая документация превращает нормальную программу в прекрасную. Ваша программа может иметь самый впечатляющий в мире набор функций, а код – лопаться от гениальнейших алгоритмов, но если никто не в состоянии пользоваться вашей программой или понять вашу работу, все это напрасно.

Поэтому на нашем уроке мы расскажем об утилитах и знаниях, необходимых для создания документации высшего качества. Даже если вы не программист, это все равно будет полезно: решив помочь какому-нибудь проекту с открытым исходным кодом, после чтения этой статьи вы сможете внести свой вклад в ее документацию. Многие проекты отчаянно нуждаются в хороших авторах для руководств, вот мы и направим вас на верный путь.

## Использование DocBook

Но прежде чем написать хоть слово, нужно выбрать формат. Большинство пользователей ныне ожидает увидеть не просто текстовые файлы README, а документацию в HTML, PDF или другие роскоши. Теоретически всю документацию можно написать в текстовых файлах и затем сделать версии в HTML и PDF, но это долго и трудоемко. И каждый раз при изменении текстовой версии придется вносить правки в версии и HTML, и PDF.

К счастью, решение есть – и это *DocBook*. Это система генерации документации из одного источника во множестве форматов (HTML, PDF, электронные книги, страницы tap и многие другие). Источник – это документ в формате XML, и работать нужно только с ним. Набор утилит преобразует файл XML в другие форматы. *DocBook* весьма популярен в сообществе сторонников свободного ПО: он применяется в KDE, Gnome и других крупных проектах.

Для тех, кто никогда не сталкивался с XML – это формат представления данных в текстовых файлах. По сути, данные хранятся в контейнерах, обозначаемых словами в угловых скобках. Представьте себе телефонную книгу с абонентами Бобом Смитом [Bob Smith] и Джо Блоггзом [Joe Bloggs]. В «обычной», текстовой версии это будет нечто вроде

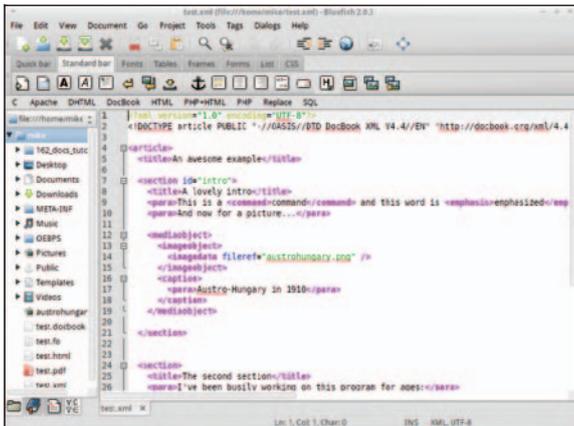
```
Name: Bob Smith
Number: 01762 271 482
Name: Joe Bloggs
Number: 08293 186 172
```

Для восприятия человеком такое хорошо подходит, но оно не слишком стандартизировано: например, в других телефонных книгах вместо двоеточий могут использоваться отступы и т. д. Без единого стандарта понадобится множество программ разборки подобных данных, и каждая будет работать по-своему. Но с помощью XML можно упростить компьютеру чтение:

```
<entry>
<name>Bob Smith</name>
<number>01762 271 482</number>
</entry>
<entry>
<name>Joe Bloggs</name>
<number>08293 186 172</number>
</entry>
```

Теперь компьютер прекрасно поймет, где начинаются и заканчиваются данные. Слова внутри угловых скобок (тэги)

# Всем поможем



» В редакторах типа *Bluefish* (на рисунке) и *Quanta* есть подсветка синтаксиса XML, это упрощает чтение кода.

не содержат данных, но идентифицируют их. В закрывающем тэге используется прямой слэш. Так, тэг `<entry>` [запись] начинает новую запись в телефонной книге, а тэг `</entry>` завершает ее (если вы знакомы с HTML, то должны понять синтаксис). Данные также можно вкладывать друг в друга, так что раздел `<name>...</name>` [имя] может быть внутри раздела `<entry>...</entry>`, как в нашем примере. Чтобы упростить чтение, можно выделять вложенные разделы отступом, но это необязательно.

Важно отметить, что XML-файлы *DocBook* фокусируются только на содержании данных, а не на их представлении. Тэги описывают только то, какую информацию они содержат – названия, параграфы и т. д. – а не то, как ее нужно отображать или выводить на печать. Это дает нам максимальную гибкость для преобразования текста во множество форматов, в чем мы вскоре и убедимся.

## Утилиты

Файлы *DocBook* можно писать в любом текстовом редакторе, но вам также понадобятся программы для их конвертации: `xsltproc` и `docbook-xsl`. Они есть в большинстве дистрибутивов, а для их установки в Ubuntu и других дистрибутивах на основе Debian достаточно вызвать `apt-get`. Откройте свой любимый текстовый редактор и введите следующий текст:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML
V4.4//EN" "http://docbook.org/xml/4.4/docbookx.dtd">
<article>
<title>Как пользоваться программой FooProg 1.0</title>
<section>
```

```
<title>Вызов из командной строки</title>
<para>Да просто наберите fooprogram!</para>
<para>Всего и делов.</para>
</section>
</article>
```

Первые две строки покажутся новичкам в XML абракадаброй, но они всего-навсего поясняют интерпретатору, что это за документ. Затем мы определяем статью [article] и ее заголовок [title] и создаем раздел [section]. Здесь вы видите множественное вложение – раздел состоит из параграфов, а статья – из разделов.

Теперь сохраните файл (в формате UTF-8) под именем `test.xml` в домашнем каталоге, откройте окно терминала и скомандуйте `xsltproc -o test.html /usr/share/xml/docbook/stylesheet/nwalsh/xhtml/docbook-xsl test.xml`

Итак, теперь у нас должен появиться файл `test.html` (если нет, и вы сделали ошибку в файле, появится сообщение об ошибке с указанием на строку). Откройте `test.html` в браузере, и вуаля – ваша документация в web-формате.

Вы видите, что команда сгенерировала даже содержание; если добавить в документ новые разделы (`<section>...</section>`), ссылки на них появятся и в содержании.

## Введение в XSLT

Теперь посмотрим, что именно делает команда. Вы помните, что в файлах *DocBook* нет никакой информации о представлении данных, а в HTML-версии у нас есть большие и маленькие шрифты, горизонтальная линия и другие элементы формата. Откуда они взялись?

Ответ – XSLT (Extensible Stylesheet Language Transformations – преобразования расширяемого языка стилей). Это система, посредством которой тэги XML преобразуются в нечто иное с добавлением информации о стиле. В нашей команде мы вызываем `xsltproc`, процессор XSLT, с нашим файлом `test.xml` в качестве входного и `test.html` в качестве выходного (задается параметром `-o`). Но файл, который совершает все волшебство – `/usr/share/xml/docbook/stylesheet/nwalsh/xhtml/docbook-xsl`

Заглянув в этот файл, вы мигом заработаете головную боль, потому что там сплошь непонятный технояз и структуры. Достаточно будет сказать, что в нем описывается, как тэги XML в файле *DocBook* конвертируются в HTML, и указывается необходимое форматирование.

Есть и другие таблицы стилей, для конвертации в другие форматы. Например, для преобразования в PDF сначала нужен промежуточный формат XSL-FO (это XML с информацией о форматировании):

```
xsltproc /usr/share/xml/docbook/stylesheet/docbook-xsl/fo/
docbook-xsl test.xml > test.fo
```

»

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

Затем нужно преобразовать **test.fo** в файл PDF. Это делается разными способами; один из самых простых – утилитой **fop** (в Debian/Ubuntu она устанавливается командой **apt-get install fop**). Затем выполните команду:

```
fop test.fo test.pdf
```

По написании документации вы сможете создавать скрипты (или даже пользоваться **makefile**) для генерации версий в HTML, PDF и любых других форматах одной командой.

Наряду с тэгами **<section>**, **<title>** и **<para>** есть другие возможности для структуризации содержимого. Например, вот так создаются маркированные списки:

```
<title>Facts</title>
<itemizedlist>
<listitem><para>Пиво - дело хорошее!</para></listitem>
<listitem><para>Да и винишко тоже.</para></listitem>
<listitem><para>Но Минздрав предупреждает.</para></listitem>
</itemizedlist>
```

Для нумерованного списка замените **itemizedlist** на **orderedlist**. У многих тэгов есть атрибуты, меняющие их поведение – например, попробуйте вместо простого **<orderedlist>** следующий:

```
<orderedlist numeration="upperroman">
```

Теперь элементы списка будут нумероваться римскими цифрами – и вместо **1, 2, 3, 4** вы увидите **I, II, III, IV** и т. д. Еще один полезный атрибут этого тэга – **continuation**. Если установить его в **continues**, то нумерация будет продолжена с предыдущего списка.

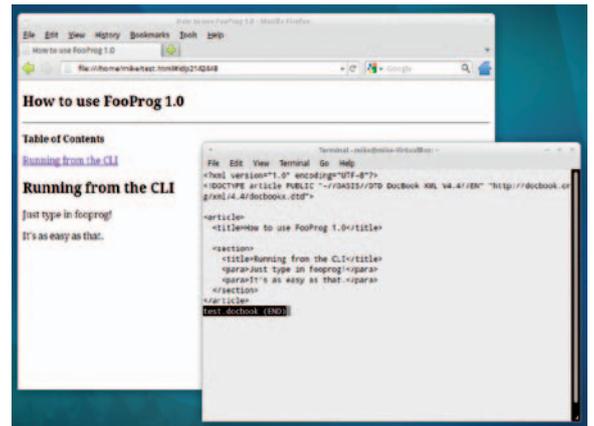
Атрибутов слишком много, чтобы описывать их подробно; если вы хотите узнать прочие возможности тэга, загляните в официальную документацию [www.docbook.org/tdg51/en/html](http://www.docbook.org/tdg51/en/html) и найдите раздел «DocBook Element Reference».

С тэгом **<programlisting>** включаются фрагменты кода:

```
<programlisting>
10 PRINT INKEY$
20 GOTO 10
</programlisting>
```

В HTML это выводится в виде моноширинного шрифта. Тэг **<blockquote>** делает отступ, а **<example>** выводит заголовок 'Example [Пример]' с номером – номера генерируются автоматически, и если вы добавите новый пример посреди документации, вам не придется перенумеровывать следующие.

Выделить раздел помогут тэги **<command>** или **<emphasis>**. (В HTML содержимое первого выделяется жирным шрифтом,



➤ Первые результаты нашего преобразования – незатейливо, но работу свою делает.

второго – курсивом). Для включения ссылки на внешний сайт воспользуйтесь тэгом **<ulink>** с параметром **url**:

```
<ulink url="http://www.linuxformat.com">The best website ever</ulink>
```

Часто бывает нужно включить в документацию внутренние ссылки – на другие разделы. Для этого воспользуйтесь атрибутом **id** тэга **<section>**, на который затем можно сослаться.

```
<section id="compiling">
<title>How to build it</title>
...
<para><link linkend="compiling">Click here</link> to read the
build instructions.</para>
```

Как видите, мы задали идентификатор раздела (**compiling**) и затем создали ссылку на этот раздел.

Чтобы вставить в документ изображение, воспользуйтесь тэгами **<mediaobject>** и **<imageobject>** следующим образом:

```
<mediaobject>
<imageobject>
<imagedata fileref="austrohungary.png" />
</imageobject>
<caption>
<para>Австро-Венгрия в 1910 г.</para>
</caption>
```

## Пять шагов к отличной документации

Хотя в этой статье мы в основном говорили о технической стороне дела, всегда стоит помнить и о человеческой стороне, т. е. о том, как сделать ваши руководства максимально доступными.

### 1 Думайте как пользователь

Представьте, что вы видите программу (или фрагмент кода) впервые. Какие вопросы вы зададите? Какие задачи вы будете решать прежде всего? Какие элементы интерфейса могут оказаться непонятными?

Важно быть осторожным и не закладывать на предыдущий опыт пользователя. Даже если это обычная программа – текстовый процессор или почтовый клиент – подумайте о том, что пользователь мог установить Linux всего несколько часов назад.

### 2 Структура прежде всего

Старайтесь максимально разбивать информацию на отдельные фрагменты, которые не должны быть слишком большими. Например, в нашем журнале есть разделы, заголовки, врезки и другие возможности, которые помогают давать информацию не в одном большом куске текста.

### 3 Не документируйте все

Были времена (особенно в мире Windows), когда маленькие знаки вопроса были разбросаны по всем окнам, чтобы вы могли понять, что делает каждая кнопка. Но вполне очевидно, что кнопка «Заккрыть» закрывает диалоговое окно. Поэтому сосредоточьтесь на основных возможностях и ключевых аспектах – нет нужды писать пять абзацев текста о том, что делает кнопка «Сохранить».

### 4 Беседуйте с разработчиками

Если вы документируете чужую программу, поддерживайте регулярную связь с разработчиками, особенно с теми, что отвечают за интерфейс. Найдя что-нибудь слишком странное на вид, чтобы быть задокументированным, или то, что можно существенно упростить, обязательно дайте знать разработчикам. Ваш совет может иметь для них большое значение.

### 5 Найдите подопытного кролика

По возможности найдите кого-то, кто никогда не пользовался программой, дайте ему свеженарисованную документацию и смотрите, как он будет работать с программой. Это прекрасный способ найти недочеты в документации – или вознаградить вас тем, как хорошо она помогает пользователю!

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

```
</mediaobject>
```

Обратите внимание на закрывающий слэш в тэге `<imagedata>` – если внутри тэга нет никаких данных, в конец нужно помещать слэш. Также в этом примере продемонстрировано, как добавляется заголовок.

В *DocBook* много других тэгов, но обычно они весьма специфичны и не подходят для общей компьютерной документации.

Тэгов, которые мы изучили, вполне достаточно для написания руководств и справочников для проектов с открытым исходным кодом.

Между версиями *DocBook* 4 и 5 есть некоторые различия; первая до сих пор широко используется и хорошо документирована в Сети – поэтому здесь мы сосредоточились на ней. Но основные знания, которые вы получили, подходят к обеим версиям.

## Написание man-страниц

Согласно правилам Debian GNU/Linux, у каждой программы должна быть man-страница. Сколько раз вам в каталоге `/usr/bin` попадалась программа со странным именем, и вы не представляли, что она делает? Даже простейшая man-страница с пятистрочным описанием программы все изменит.

Для графической программы, функциональность которой проявляется внутри программы, а не через параметры командной строки, подробная man-страница не нужна. Но для утилит командной строки хорошая документация имеет решающее значение. Для доступа к man-страницам используется одноименная команда – например, `man ls`. Эти страницы пишутся не простым текстом, а особым языком разметки с дополнительным форматированием. Просмотреть страницу с разметкой можно, например, так:

```
zless /usr/share/man/man1/ls.1.gz
```

Если вы хотите поработать с текстовыми файлами, можете взять небольшую страницу из этого каталога (например, `znew.1.gz`), скопировать ее в свой домашний каталог, распаковать ее командой `gunzip` и поработать с ней. Чтобы просмотреть ее программой `man`, укажите путь и имя файла:

```
man ./znew.1
```

(Набрав просто `man znew`, вы увидите версию из `/usr/share/man/man1`.) Рассмотрим несколько первых строк этого файла:

```
.TH ZNEW 1
```

```
.SH NAME
```

```
znew \- преобразует файлы .Z в файлы .gz
```

```
.SH SYNOPSIS
```

```
.B znew
```

```
[ -ftv9PK ] [ name.Z ... ]
```

```
.SH DESCRIPTION
```

```
.I Znew
```

```
преобразует файлы из формата .Z (compress) в формата .gz (gzip).
```

Если вы хотите заново сжать файл в формате `gzip`, переименуйте файл:

чтобы он получил расширение `.Z`, затем примените `znew`.

```
.SH OPTIONS
```

```
.TP
```

```
.B \-f
```

Преобразование из формата `.Z` в формат `.gz`, даже если файл `.gz` уже существует.

В языке разметки man-страниц команды форматирования начинаются с точки. Итак, прямо в первой строке мы видим `.TH`, что означает «заголовок страницы [title header]». За ним следуют имя программы и род man-страниц, к которой она принадлежит (1 для пользовательских команд, 5 для конфигурационных файлов и 8 для утилит системного администрирования). Здесь также можно указать дату в кавычках, например, “23 September 2012”, и она появится в верхней части man-страницы. Затем следует `.SH`, заголовок раздела [section header]. Он выводится жирным шрифтом, выровненным по левому краю окна, обычно заглавными буквами. Полная, хорошо написанная страница man должна иметь следующие разделы:

» **NAME** Описание в одну строку

» **SYNOPSIS** Параметры команды

» **DESCRIPTION** Краткое пояснение работы программы

» **OPTIONS** Подробное описание параметров

» **EXAMPLES** Примеры решения различных задач (обязателен, если у программы много параметров)

» **BUGS** Любое необычное поведение программы

» **AUTHOR** Ваш электронный адрес и/или ссылка на сайт

Опции `.B` и `.I` выделяют текст жирным шрифтом и курсивом соответственно. На обычных терминалах Unix курсив не отображается, поэтому текст с опцией `.I` может подчеркиваться или выделяться другим цветом. Вы видите, что `.TP` начинает новый абзац с отступом для всех строк, кроме первой. Тире лучше предварять обратным слэшем (`\-`), иначе процессор может интерпретировать их как дефисы.

Если кому по душе графические программы, здесь вам крупно не повезло. Программа *ManEdit* неплохо справлялась с этой задачей, но она долго не обновлялась, и ее сложно заставить работать в приличных дистрибутивах (она основана на *GTK 1*). В *Gmanedit* есть пошаговый мастер создания новой man-страницы, но вам все равно придется работать с размеченным текстом. **LXF**

### Скорая помощь

Нужно вставить угловые скобки `>` и `<?` Если ввести их как есть, это собьет с толку парсер, поэтому пользуйтесь обозначениями `&gt;`; (`greater than` – больше чем) и `&lt;`; (`less than` – меньше чем). Тогда парсер поймет, что это специальные символы, не имеющие ничего общего с тэгами.

## Работа с документами KDE и Gnome

Оба главных рабочих стола используют для документации *DocBook*, и обоим проектам всегда нужны новые помощники. Самый простой способ помочь – запустить свои любимые программы Gnome/KDE, открыть окно справки и посмотреть, чего не хватает. Некоторые программы уже имеют обширную документацию, но ее все равно можно улучшить – например, она может быть неактуальной или в ней может не хватать раздела. Найдя функцию, которую трудно понять, загляните в справку, и если она недостаточно хорошо описана, вы можете это улучшить.

Важный совет: если вы найдете программу без документации (или с очень плохой документацией), не хватайтесь за дело сразу и не пишите сотни слов. Сначала свяжитесь с разработчиками или найдите IRC-канал проекта и узнайте, не работает ли над документацией кто-то еще.

Будет досадно, если вы потратите несколько недель на большой фрагмент текста и окажется, что готовое описание уже есть!

В Gnome имеется удобное руководство для новых авторов – <http://developer.gnome.org/gdp-handbook/>

[stable/gettingstarted.html.en](http://stable/gettingstarted.html.en), эквивалент для KDE – <http://community.kde.org/GetInvolved/documentation>.

На каждой из этих страниц вы найдете ссылки на другие руководства, которые объясняют приемы и соглашения, принятые проектами. Например, для KDE существует руководство по стилям (<http://l10n.kde.org/docs/styleguide/index.html>), объясняющее использование аббревиатур, номер версии и дат; кроме того, там содержатся советы общего характера по написанию хорошей документации. Пренебрегать ими будет себе же во вред.



# Arduino: Плюс

Нужны устройства ввода? Ник Вейч знакомит нас со сканируемой матрицей и рассказывает о вариантах подключения кнопок к Arduino.



## Наш эксперт

Когда LXF только появился, его держали на плаву исключительно скрипты Bash от Ника Вейча. Потом их заменили «люди», и это, по мнению Ника, стало шагом назад...

## Что нам надо

- » ПО Arduino 1.0
- » Несколько кнопок
- » Столько же импульсных диодов
- » Сдвиговый регистр SIPO
- » Сдвиговый регистр PISO
- » Монтажная плата, много разъемов и проводников

**Н**а этом уроке мы поговорим о матрице. Нет, не той, что с красными и синими пилюльками, а о клавиатуре. Раньше мы уже кратко о ней упоминали, а сегодня детально познакомимся с вариантами и стратегией работы с ними. Начнем с более простого.

Как всегда, способов подключения кнопок к Arduino несколько. В самом простом случае одна кнопка назначается на один контакт. Если кнопок не больше четырех, это и вообще лучший вариант: тут вы все равно мало что сэкономите. Если кнопок больше, придется подумать о матрице, или о некотором уплотнении данных.

## Методы уплотнения

Простое уплотнение можно реализовать с помощью различных устройств – например, есть специальные микросхемы, сворачивающие восемь линий данных в одну, используя 3 бита как адрес. Самый популярный мультиплексор/демультиплексор – микросхема 4051 и ее варианты; все они работают по описанному выше принципу. Поскольку это всего лишь переключатель, он может работать в обоих направлениях для цифровых и аналоговых входов и выходов, но подходит и для создания клавиатуры.

Схема будет довольно простой (см. Схему 1). При восьми переключателях довольно легко перебрать значения на выводах адреса и считать выходное значение. Единственная сложность по части кода – это задание значений на адресных выводах. К счастью, есть удобная функция `bitread(x,y)`, которая возвращает значение бита для заданного целого числа `x`. Для перебора значений нам понадобится нечто вроде

```
int data;
for (int count=0; count<=7; count++) {
  digitalWrite(4, bitRead(count,0));
  digitalWrite(5, bitRead(count,1));
  digitalWrite(6, bitRead(count,2));
  bitWrite(data, count, digitalRead(7));
}
```

Функция `bitWrite()` собирает отдельные биты в одно целое значение, а каждый вход соответствует одному биту результирующего значения. Здесь предполагается, что 4, 5 и 6 – адресные биты, а 7 – входной контакт.

Этого способа вполне достаточно, чтобы читать данные с простой клавишной панели, не прибегая к реализации полной матрицы. Ограничения этого подхода проявляются при большем количестве клавиш. Можно, конечно, воспользоваться мультиплексорами с большим количеством входов, но если их действительно много, схема может стать довольно неуклюжей и медленной. В этих случаях нам понадобится матрица...

## Строки и столбцы

В матрице клавиши организованы по строкам и столбцам. Как и в схеме с мультиплексором, кнопки нужно периодически сканировать, чтобы проверить, нажаты они или нет, но в случае с матрицей это обычно делается группами. По сигнальному проводнику на каждый столбец поступает ток, затем любые нажатые клавиши передают этот ток в проводники соответствующих строк, значения которых можно считать по отдельности или все сразу.

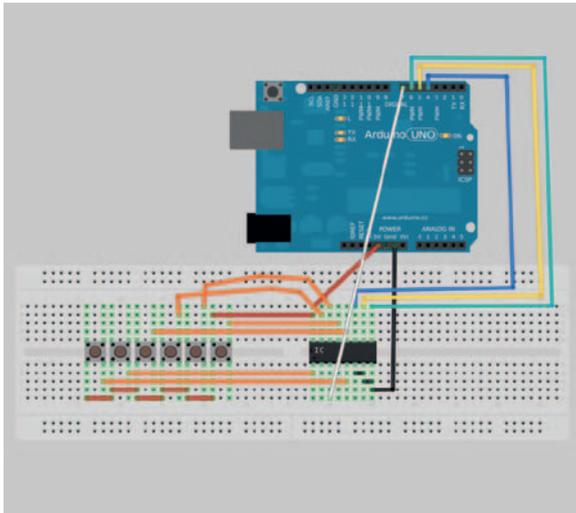
Как получить этот сигнал? Вариантов опять-таки несколько, в том числе – просто запитать линии напрямую с выходов (см. врезку о маленьких матрицах). Мы воспользуемся сдвиговым регистром, который легко настроить для этой цели (см. Схему 2).

Функциями сдвигового регистра, который формирует ток для столбцов, нужно управлять на более низком уровне – сдвиговые регистры в Arduino перемещают по байту данных за цикл, но мы воспользуемся регистром на уровне битов – загрузим бит в регистр и затем с помощью импульсов будем сдвигать его на одну позицию за раз. Глобальная переменная `glob_COLUMN` содержит номер текущего столбца.

```
glob_COLUMN++;
digitalWrite(SR_LATCH, LOW);
if (glob_COLUMN>NUMBER_OF_COLUMNS) {
  glob_COLUMN=0;
  digitalWrite(SR_DATA, HIGH); //заново загружаем 1 на вывод
  digitalWrite(SR_CLOCK, HIGH); //сдвигаем
  digitalWrite(SR_CLOCK, LOW);
  digitalWrite(SR_DATA, LOW); // загружаем 0 на вывод
}else{
  digitalWrite(SR_CLOCK, HIGH);
```

» Месяц назад Мы превратили Arduino в чат-бота.

# Клавиатура



➤ Небольшой набор кнопок легко обработать мультиплексором/демультиплексором.

```
digitalWrite(SR_CLOCK, LOW);
}
digitalWrite(SR_LATCH, HIGH);
```

Переменные **SR\_** соответствуют контактам Arduino, используемым для подключения к этим сигналам на сдвиговом регистре. Никаких соглашений на этот счет нет – можно использовать любые контакты, но лучше объявить их в начале кода, чтобы при необходимости легко менять в зависимости от того, какие контакты используются/свободны.

Пока счетчик столбцов не достиг количества столбцов в нашей матрице, код лишь устанавливает импульс в «единицу», а затем снова в «ноль». Частотная характеристика стандартного сдвигового регистра **59581** с запасом покрывает последовательные вызовы **digitalWrite()**, но если вы будете пользоваться другими компонентами или если в цепи большой шум, может потребоваться небольшая задержка между вызовами. В большинстве случаев будет достаточно такой:

```
delayMicroseconds(5);
```

Когда мы достигаем количества столбцов, срабатывает условие **if**. Оно сбрасывает счетчик столбцов и выдает еще одну «единицу» на выходной контакт, затем сдвигает его, и цикл начинается снова. Так, для трех столбцов последовательность будет такой: **100100100...** а для четырех – такой: **100010001...** Разумеется, можно объединить несколько регистров сдвига в цепочку и использовать большие значения счетчиков для матрицы.

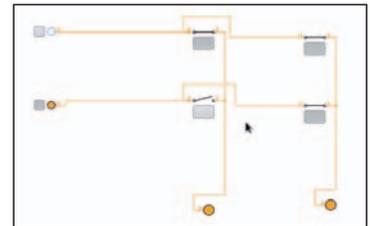
При подаче сигнала на каждый из столбцов нужно просканировать строки. Сделать это можно по-разному – с помощью мультиплексора (как мы сделали это ранее), расширителя входа/выхода или сдвигового регистра другого типа.

Расширитель входа/выхода (например, **MCP28S17**) хорошо использовать в том случае, если матрица большая. Эта микросхема использует последовательный интерфейс (SPI), для которого

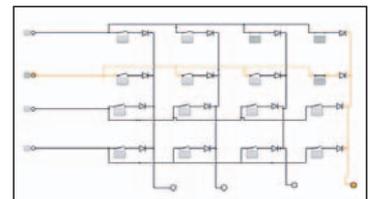
## Ложное срабатывание и маскирование

У некоторых клавиатур на основе сканируемых матриц есть одна проблема – ложное срабатывание. При нажатии нескольких клавиш сканирующей цепи кажется, что нажата совершенно другая клавиша. Если клавиши находятся в разных строках и столбцах, то соединения, которые они формируют, позволяют току течь в обоих направлениях. В некоторых случаях, например, в углах матрицы, это может приводить к тому, что ток, обычно протекающий через один выключатель и через другой в противоположном направлении, дает ложный сигнал.

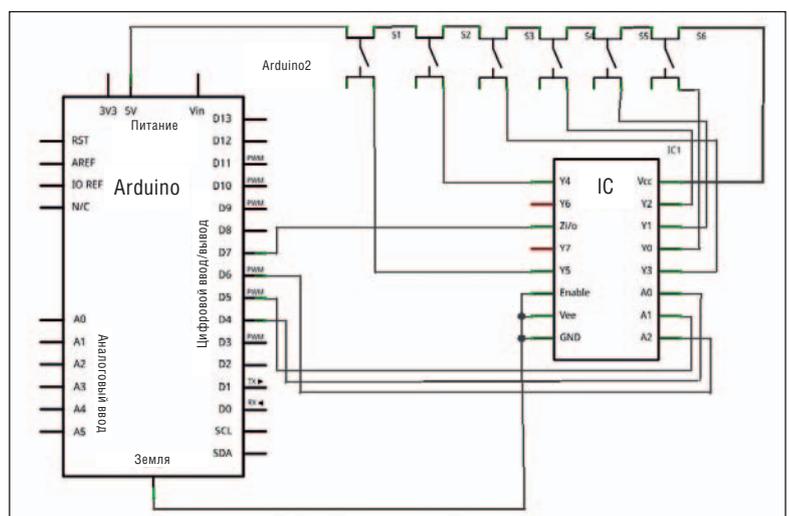
Обычно это не проблема, кроме тех случаев, когда вам нужно прочитать несколько одновременных нажатий клавиш. Решение в этом случае тоже довольно простое, хотя и несколько трудоемкое – диоды. Включение диода последовательно с каждой клавишей не позволит сигналу пройти обратно через закрытый выключатель и снимет проблему ложного срабатывания. Есть и другой вариант, используемый в библиотечной матрице клавиатуры Arduino: опрашивать только одну строку и один столбец в один момент времени. По различным причинам, описанным в основном тексте, это решение плохо масштабируется.



➤ Ложное срабатывание возникает потому, что несколько нажатых клавиш позволяют сканирующему току «утечь» из соответствующих строки и столбца и найти обходной путь.



➤ Диоды предотвращают утечку тока. Они очень дешевы, и их удобно использовать для подключения к матрице на плате или на макете, поэтому веских причин избегать их нет (падение напряжения на них недостаточно велико, чтобы помешать правильному считыванию сигнала).



➤ Схема 1. Мультиплексор поможет определить состояние нескольких переключателей, не используя слишком много входов. Это несложно и с точки зрения программирования.

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

## Скорая помощь



Матрица клавиатуры не обязана быть квадратной. Например, с точки зрения адресации матрица может быть размером 8×8, а физически она может быть любой формы.

➤ **Схема 2. Более крупная матрица на сдвиговых регистрах — обязательно проверьте все соединения тестером!**

нужно 4 контакта, но зато он очень быстр — помните, что с разрастанием матрицы время на ее сканирование увеличивается, поэтому если нужно и обрабатывать данные, и считывать значения с клавиатуры, сканирование должно быть быстрым.

Впрочем, в качестве расширителя проще всего использовать еще один сдвиговый регистр. Ранее в этой серии мы применяли сдвиговые регистры SIPO (с последовательным входом и параллельным выходом), на которые подается последовательность битов, выводимых на выходные контакты. Есть регистры и с параллельными входами и последовательным выходом (PISO), такие как **74165** и **CD4021**. В таких схемах значения восьми входов снимаются и выдаются в виде последовательности на один выход под управлением сигнала внешнего генератора — по сути, их действие противоположно регистрам SIPO. Как и другие сдвиговые регистры, их тоже можно соединять в цепочку, и мы сможем получить данные в строках с числом элементов больше 8. Для их работы нужно всего три контакта.

Вот каким образом сканируется строка:

```
digitalWrite(SRIN_LATCH, HIGH);
delayMicroseconds(25);
digitalWrite(SRIN_LATCH, LOW);
data = shiftIn(SRIN_DATA, SRIN_CLOCK);
```

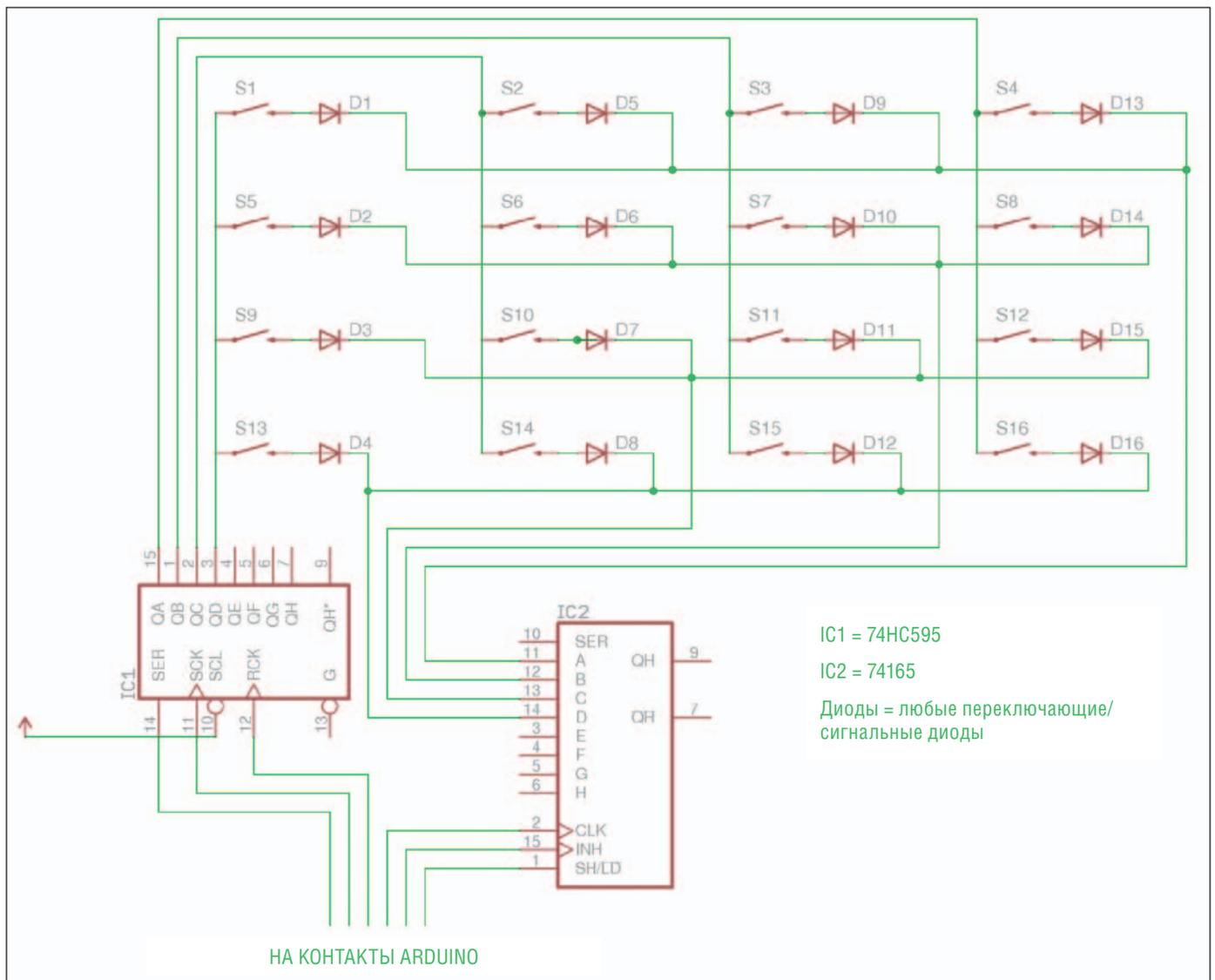
## Немного о Define

Директива **#define** очень полезна: стоит добавить несколько таких в начало кода — и при необходимости можно легко изменить какую-то функцию без необходимости просматривать весь код. Это, в отличие от переменных, также экономит память. Замена значений в коде на объявление производится перед компиляцией и равноценно указанию самих значений. Директивы лучше размещать в начале кода. Запомните синтаксис! Без знака + и без ; в конце:

```
#define MYVARIABLE 42
```

Имена параметров в таких директивах обычно пишутся заглавными буквами — так их проще обнаружить в коде. Кроме того, не бойтесь выполнять над ними арифметические операции — они тоже выполняются прекомпилятором.

Перевод контакта-«защелки» в «единицу» функцией **digitalWrite()** «загрузит» вход «защелки». Небольшая задержка поможет значениям установиться, прежде чем мы снова переведем «защелку» в ноль для «заморозки» входа.



➤ **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.

После этого нужно лишь воспользоваться функцией `shiftIn()` стандартной библиотеки Arduino для фиксации значения этого байта. Если сдвиговой регистр не один, то входы «защелки» объединяются вместе, и понадобится еще один вызов `shiftIn()` для чтения дополнительного байта данных.

## Борьба с дребезгом контактов

Одна из проблем с физическими переключателями, будь то мембранные клавиатуры, тактильные кнопки или переключатели с самовозвратом любых типов, заключается в том, что сигнал может «дрожать». Это термин, используемый для описания эффекта физической природы самих контактов, которые включены и выключены не четко. Такое бывает из-за физической вибрации контакта, поэтому, например, при отпускании клавиши контакты могут дрогнуть и снова замкнуть цепь, пусть и ненадолго.

Если вы хотите отфильтровать подобные нежелательные события, в принципе это делается на уровне электроники, но тут могут быть проблемы, так что позаботимся об этом на уровне программы. Довольно простой и ясный способ – просто объединить входящие данные с последним набором данных, полученных из матрицы, по схеме И.

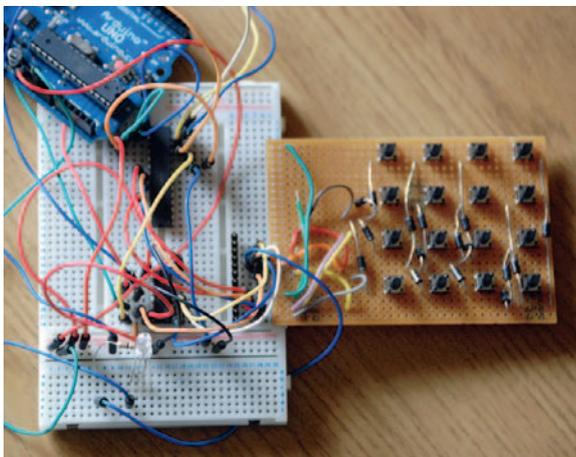
```
data &= shiftIn(SRIN_DATA, SRIN_CLOCK);
```

Время между сканированием двух наборов данных (если вы не сканируете быстро – в этом случае может использоваться несколько буферов) должно быть достаточным для того, чтобы состояние клавиш установилось. Конечно, оно должно быть и достаточно коротким, чтобы избежать ситуации, когда быстро нажатая клавиша не распознается. По общей рекомендации, время между операциями сканирования должно быть менее 200 мс.

## Объединяем все вместе

На практике функция чтения клавиатуры скорее всего реализуется как сервис прерываний. В этом случае нужно правильно задать таймер (см. LXF155), чтобы прерывание выдавалось достаточно регулярно для перехвата нажатий клавиш, но не слишком часто, чтобы не съедать все процессорное время. Оно зависит от размера матрицы и метода сканирования – помните, что с ростом размера матрицы время на ее сканирование увеличивается по квадратичной функции. В примере кода на DVD есть процедура прерывания для матрицы 4×4 с использованием сдвиговых регистров для управления строками и столбцами.

В зависимости от нажатия клавиш вам также нужно выполнять соответствующие действия и, конечно, помещать данные туда,

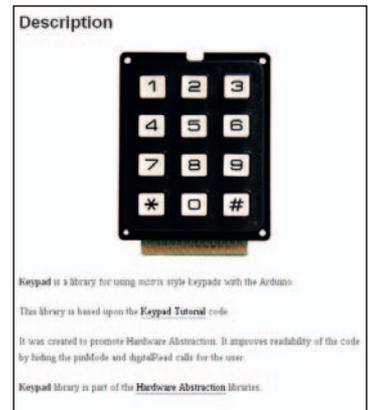


» На макете удобно создавать прототипы объектов, так как можно подключать требуемые полупроводники к Arduino.

## Маленькие матрицы

В ПО Arduino есть библиотека матрицы клавиатуры, которая сканирует матрицу клавиш, не вызывает блокировки и обладает зачаточной обработкой событий. Она работает с различными клавиатурами и использует простой режим сканирования, исключая эффект ложного срабатывания на клавиатурах без диодов.

Почему мы не воспользовались ею? Единственная проблема в том, что для нее нужно по контакту на каждую строку и столбец, т.е. для матрицы 4×4 нужно восемь контактов – заметная часть выводов Arduino. Использование сдвиговых регистров (в основном коде) позволяет снизить их количество до шести, а при объединении нескольких сдвиговых регистров можно работать с гораздо большими матрицами.



» Если у вас маленькая матрица и вам не жаль контактов Arduino, воспользуйтесь отличной библиотекой в ПО Arduino.

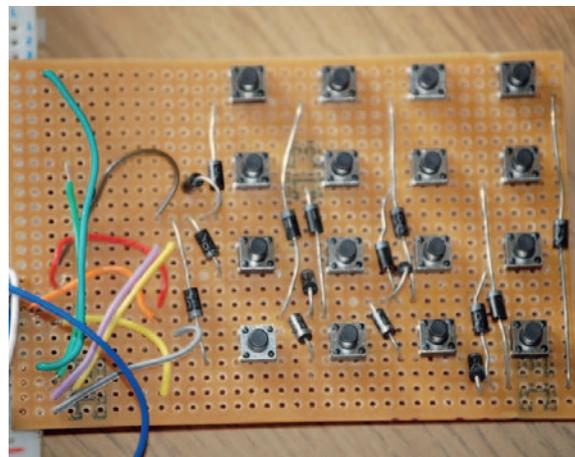
где главная программа их найдет. Назначение клавиш зависит от вашего приложения, но для хранения нажатий клавиш удобно пользоваться буфером `buffer` и счетчиком `count`. В основной программе можно проверять счетчик, чтобы узнать, были ли нажаты клавиши.

```
if (data > 0){
  if (buffercount < BUFFER_LIMIT){
    buffercount++;
    buffer[buffercount] = data;
  }
}
```

Тщательно подобрав соответствующие компоненты, легко просканировать матрицу 16×16 без особых затрат процессорного времени.

matrix

Макет – не самое замечательное в мире место для сборки матричной клавиатуры: уж слишком много на нем соединений. Проще будет взять монтажную плату и разместить клавиши на ней (в тех местах, где это необходимо, дорожки надо будет перерезать). LXF



» Клавиатуру удобнее монтировать на монтажной плате. Но часть дорожек придется перерезать – тут помогут инструменты Dremel!

## Скорая помощь

Хотя многие компоненты довольно типичны – один регистр SIPO почти ничем не отличается от другого – всегда заглядывайте в документацию, чтобы проверить соответствие уровней напряжения и быстродействия вашей схеме.

» **Через месяц** Матрица служит не только для ввода – устроим световое шоу.



# Сканеры: hping3,

В развитие темы сетевого сканера, затронутой в прошлом номере, Игорь Штомпель рассматривает его графические оболочки.



## Наш эксперт

**Игорь Штомпель** каждый день открывает в Linux что-то новое и использует его где только возможно. А Unity ему в этом ничуть не мешает.



На прошлом уроке мы занимались важным инструментом обеспечения сетевой безопасности – *Nmap*. Но это инструмент командной строки, в которой далеко не все чувствуют себя комфортно. И, конечно же, разработчики не обошли вниманием создание графических интерфейсов для *Nmap*. О них мы сегодня и поговорим.

## hping3: обзаведемся

*hping3* (<http://www.hping.org/>) – это сканер портов, ориентированный на использование в командной строке. Программа позволяет генерировать различные пакеты (TCP, UDP, ICMP и др.). Общее назначение *hping3* – отправка на сканируемый хост TCP/IP-пакетов,

► Рис. 1. SYN-сканирование с помощью *hping3*.

```
igor@make02:~$ sudo hping3 -I eth0 -S scanme.nmap.org -p 80
HPING scanme.nmap.org (eth0 74.207.244.221): S set, 40 headers + 0 data bytes
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=0 win=14600 rtt=304.4 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=1 win=14600 rtt=324.6 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=2 win=14600 rtt=314.1 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=3 win=14600 rtt=295.3 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=4 win=14600 rtt=297.7 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=5 win=14600 rtt=290.3 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=6 win=14600 rtt=305.5 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=7 win=14600 rtt=289.6 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=8 win=14600 rtt=297.2 ms
len=46 ip=74.207.244.221 ttl=52 DF id=0 sport=80 flags=SA seq=9 win=14600 rtt=283.3 ms
^C
--- scanme.nmap.org hping statistic ---
11 packets transmitted, 10 packets received, 10% packet loss
round-trip min/avg/max = 283.3/308.7/324.6 ms
```

заголовки которых могут модифицироваться с помощью ключей соответствующей команды (о них мы поговорим ниже). Благодаря гибкой реализации этой функциональности возможно осуществление разнообразных видов сетевого сканирования.

Автором *hping3* является Сальваторе Санфилиппо [Salvatore Sanfilippo], который выдал ее в свободный доступ на условиях GNU GPLv2 (отдельные компоненты – на условиях лицензии BSD, см. файл `copyright` в каталоге `/usr/share/doc/hping3`). *Hping3*, как и *Nmap*, является кросс-платформенным решением, но доступным для меньшего количества операционных систем по сравнению с *Nmap*: это Linux, семейства Unix, Mac OS X, Windows.

В Ubuntu 12.04 *hping3* устанавливается следующим образом:

```
$ sudo apt-get install hping3
```

Для этих целей можно также воспользоваться *Synaptic* (недоступен в Ubuntu 12.04 по умолчанию) или Центром приложений Ubuntu.

*hping3* имеет ряд ключей, которые позволяют повлиять на тип сканирования (специфицировать соответствующие флаги – управляющие биты). Среди них:

- » **-S** (или **--syn**) – флаг SYN (информирует о начале TCP-соединения);
- » **-P** (**--push**) – флаг PUSH (передать данные из TCP-пакета в соответствующее прикладное программное обеспечение);
- » **-R** (**--rst**) – флаг RST (запрос на установление нового соединения (вместо неудавшегося), сам флаг устанавливает в ответ на получение неверного TCP-пакета);
- » **-F** (**--fin**) – флаг FIN (запрос на закрытие соединения, признак конца потока данных, необходимо подтверждение от принимающей стороны);
- » **-U** (**--urg**) – флаг URG (TCP-пакет содержит важные данные);
- » **-A** (**--ack**) – флаг ACK (предыдущий пакет был получен).

Рассмотрим примеры сканирования с использованием некоторых из приведенных выше ключей. Дадим следующую команду:

```
$ sudo hping3 -I eth0 -F scanme.nmap.org -p 80
```

Вот что в ней происходит:

- » **-I eth0** – мы специфицируем интерфейс с помощью которого будет осуществляться сканирование;
- » **-F** – ключ *hping3* (специфицирует флаг FIN генерируемых TCP-пакетов);
- » **scanme.nmap.org** – цель сканирования;
- » **-p 80** – порт назначения (доступность которого исследуется).

Как показано на рис. 1, в выводе, среди прочего, мы видим, что `flags` имеет значение “SA”. Это означает, что порт открыт (в противном случае значение было бы “RA”). В данном случае, S говорит о флаге SYN, а A – о флаге ACK, т.е. что отправленный пакет был получен. Более подробную информацию о формате вывода *hping3* для TCP можно получить с помощью страниц руководства (см. раздел “TCP OUTPUT FORMAT”):

```
$ man hping3
```

Изменим номер сканируемого порта, например, на 81:

```
$ sudo hping3 -I eth0 -F scanme.nmap.org -p 81
```

Теперь в выводе *hping3* мы увидим, что значение `flags` стало равно “RA”, что означает: порт закрыт. Кроме `flags`, обратите

# Zenmap и Umit

внимание и на `sport`. Последнее говорит нам о том, какой порт сканировался. Данное значение нам пригодится на следующем шаге.

Запустим сканирование с помощью `hping3`, при котором будет отправляться по одному пакету на каждый порт, начиная с 80, увеличиваясь с шагом 1 (т.е. отправили пакет на 80, следующий пакет – на 81 и т.д.):

```
$ sudo hping3 -S -p ++80 scanme.nmap.org
```

Как видно в выводе этой команды, значение `sport` каждый раз увеличивается на один, а значение `flags` говорит нам о том, закрыт или открыт тот или иной порт.

С помощью ключа `-c` (`--count`) можно специфицировать количество получаемых пакетов:

```
$ sudo hping3 -S -c 4 -p ++80 scanme.nmap.org
```

В данном случае, после получения четырех ответных пакетов выполнение команды будет прекращено.

По умолчанию `hping3` использует протокол TCP. Другие протоколы можно указать с помощью специальных ключей:

- » **-0** (`--rawip`) – работать в режиме RAW IP;
- » **-1** (`--icmp`) – работать в режиме ICMP;
- » **-2** (`--udp`) – работать в режиме UDP;
- » **-8** (`--scan`) – работать в режиме сканирования (необходимо указать диапазон портов и тип сканирования);
- » **-9** (`--listen`) – работать в режиме прослушивания (необходимо указать сигнатуру, на предмет которой будет осуществляться прослушивание).

Работа в первых трех режимах, думаю, затруднений не вызовет. Приведем пример:

```
$ sudo hping3 -I eth0 -1 -c 10 scanme.nmap.org
```

Здесь мы задействуем протокол ICMP и получаем десять соответствующих пакетов.

А вот использование двух последних режимов представляет больший интерес. Начнем с режима сканирования:

```
$ sudo hping3 -I eth0 -8 80-85 -S scanme.nmap.org
```

В данном случае мы осуществляем SYN-сканирование (ключ `-S`) портов с 80 по 85 на [scanme.nmap.org](http://scanme.nmap.org). В выводе этой команды, как показано на рис. 2, мы видим таблицу, в которой отображаются номер порта, название сервиса, флаги (открыт/закрыт порт) и т.д. Из вывода нам становится понятно, что из шести просканированных портов открыт только 80.

Допускается указывать порты для сканирования не только с помощью диапазона, но и через перечисление. Скажем, так:

```
$ sudo hping3 -I eth0 -8 22,80,9929 -S scanme.nmap.org
```

Режим прослушивания позволяет отслеживать приходящие пакеты на предмет наличия в них указанной сигнатуры. Например, отследим наличие в пакетах `Nmap`:

```
$ sudo hping3 -I eth0 -9 nmap scanme.nmap.org
```

Дайте эту команду, затем запустите браузер и откройте сайт [scanme.nmap.org](http://scanme.nmap.org). В выводе команды вы увидите все, что соответствует указанной вами сигнатуре.

```
igor@make02:~$ sudo hping3 -8 80-85 -S scanme.nmap.org
Scanning scanme.nmap.org (74.207.244.221), port 80-85
6 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+-----+-----+
|port| serv name | flags | ttl | id | win | len |
+-----+-----+-----+-----+-----+-----+
  80 http      : .S..A... 52   0 14600 46
All replies received. Done.
Not responding ports:
```

» Рис. 2. Сканирование портов с помощью `hping3`.

Итак, знакомство с утилитой `hping3` состоялось; перейдем к рассмотрению работы с официальным графическим интерфейсом для `Nmap` – `Zenmap`.

## Оболочка Zenmap

`Zenmap` – это официальный графический интерфейс для `Nmap`, который написан на языке программирования Python и доступен на условиях GNU GPL.

В Ubuntu 12.04 `Zenmap` можно найти здесь: Интернет > `Zenmap`. Обратите внимание, что доступны две возможности запуска `Zenmap`: с правами текущего пользователя – `Zenmap` и с правами суперпользователя – `Zenmap (as root)`. При выборе первого варианта часть возможностей `Nmap` не будет доступна. Запустите `Nmap` с правами суперпользователя.

После успешного запуска `Zenmap` вы увидите главное окно программы. Интерфейс ее интуитивно понятен.

Для примера, осуществим сканирование с помощью `Zenmap`. В поле Цель введите

```
scanme.nmap.org
```

Поле Профиль пусть пока останется пустым. Далее отредактируйте поле Команда, чтобы у вас получилось следующее:

```
nmap -sP scanme.nmap.org
```

Именно с этой команды мы начинали рассмотрение работы с `Nmap` в консоли. Затем для запуска сканирования нажмите кнопку Сканирование. Мы получили тот же вывод, что и при аналогичном сканировании в консоли с помощью `Nmap`.

Для сохранения отчета о сканировании можно воспользоваться меню (Сканирование > Сохранить отчет о сканировании, или `Ctrl+S`).

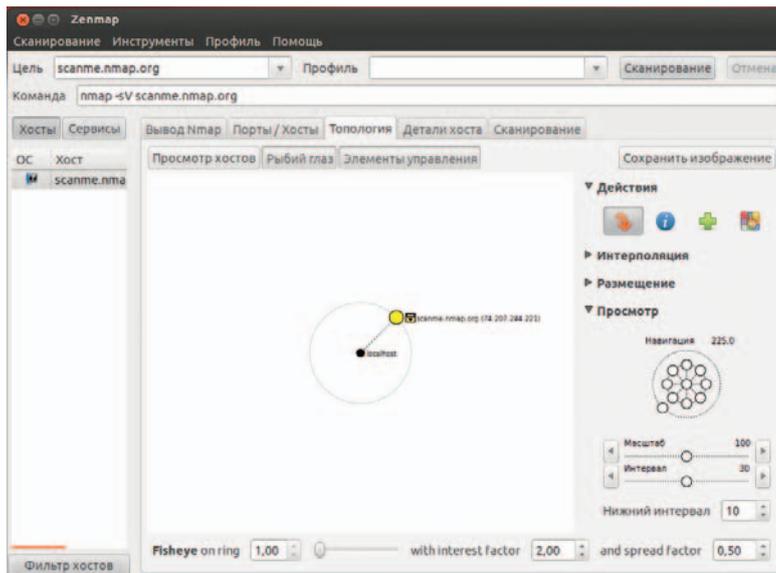
Обратите внимание, что в левой части окна есть две вкладки: Хосты, где будут отображаться цели наших сканирований (кстати, там появился хост `scanme.nmap.org` – 74.207.244.221), и Сервисы, где будут отображаться сервисы, обнаруженные на целях сканирования.

Теперь повторим сканирование, позволяющее определить версии сервисов на сканируемом хосте, но уже в `Zenmap`. Для этого отредактируйте значение в поле Команда, чтобы оно приняло уже знакомый нам вид:

```
nmap -sV scanme.nmap.org
```

Как вы поняли, вывод `Zenmap` не отличается от вывода аналогичной команды в консоли при использовании `Nmap`, но в данном случае нас интересует другое. Нажмите кнопку Сервисы. »

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



► **Рис. 3. Zenmap** отображает топологию по результатам сканирования.

Вы увидите все сервисы, которые удалось обнаружить *Zenmap*. Далее, в перечне Сервисы выберите какой-нибудь из них – и в правой части окна программы на вкладке Порты/Хосты отобразится информация о нем.

*Zenmap* позволяет наглядно представить топологию, связывающую сканирующий и сканируемые хосты. Для просмотра топологии после завершения сканирования достаточно будет перейти на вкладку Топология в правой части окна *Zenmap* (см. рис. 3).

Обратите внимание, что вы можете управлять отображением топологии с помощью кнопок вкладки Топология: Рыбий глаз и Элементы управления, а также просмотреть подробную и наглядную информацию по хостам, которые сканировались с помощью кнопки Просмотр хостов.

Вкладка Детали хоста предоставляет информацию о статусе хоста (состояние, открытые порты, отфильтрованные порты и др.), адресах (IPv4, IPv6, MAC), имени хоста.

Последняя вкладка, Сканирование, позволяет перейти к результатам любого сканирования, осуществленного в рамках текущей сессии *Zenmap*.

Но, на наш взгляд, одной из важнейших возможностей *Zenmap* является поддержка работы с профилями, о которых мы вскользь упоминали ранее. Чтобы отредактировать текущий или создать новый профиль, необходимо воспользоваться меню Профиль > Новый профиль или команда, или Редактировать выбранный профиль. Можно воспользоваться горячими клавишами – Ctrl+P или Ctrl+E соответственно.

Для примера, создадим новый профиль. Появится окно Редактор профиля, которое, к сожалению, не полностью переведено на русский язык. В окне имеется несколько вкладок (Профиль, Сканирование, Ping и т. д.). Начнем со ввода имени профиля в соответствующем поле:

**Murprof**

Далее перейдем на вкладку Сканирование. Здесь доступно для выбора большое количество опций. Выбор какой-либо из этих опций приводит к изменению команды в поле вверху окна.

В правой части окна Редактор профиля имеется поле, которое называется Помощь. В этом поле при наведении на него курсора мыши отображаются описания той или иной опции. Для опций, доступных в выпадающих списках, эта возможность *Zenmap* не актуальна.

Завершим создание нашего профиля. На вкладке Сканирование мы выбрали

Version detection (-sV)

а на вкладке Ping (SYN-пингование) –

SYN ping (-PS)

В итоге у нас получилась следующая команда:

**nmap -sV -PS scanme.nmap.org**

Для сохранения профиля необходимо нажать кнопку Сохранить изменения. Окно редактирования профиля закроется. Теперь в главном окне *Zenmap* в поле Профиль выберите из выпадающего списка Murprof и нажмите кнопку Сканирование, после чего увидите соответствующий вывод.

Кроме того, использовать окно редактирования профиля можно для изучения *Nmap*. Например, можно получить краткое описание опций сканирования для ознакомления с ними, поэкспериментировать с их выбором (как правило, если опции не могут быть использованы совместно, при выборе «конфликтующей» опции изменение команды происходить не будет). Далее, на вкладке Scripting вы сможете ознакомиться со скриптами *Nmap*. При установлении курсора на тот или иной скрипт в середине вкладки вы получите о нем подробную информацию – в том числе, как скрипт влияет на вывод *Nmap*. При выборе скрипта происходит изменение команды, что важно для первоначального ознакомления с использованием скриптов. В нижней части середины окна доступна информация об аргументах скрипта – выбрав какой-либо из них в правой части вкладки (Помощь), вы получите его краткое описание.

*Zenmap* — не единственный графический интерфейс *Nmap*.

## Еще один интерфейс — Umit

*Umit* (<http://www.umatproject.org>) – это графический интерфейс для сетевого сканера *Nmap*, доступный на условиях GNU GPLv2 (ряд файлов доступен на условиях других лицензий – см. файл **copyright** в каталоге `/usr/share/doc/umit`).

Установить *Umit* в Ubuntu 12.04 можно с помощью следующей команды:

**\$ sudo apt-get install umit**

После этого программа запускается просто-напросто командой

**\$ umit**

При каждом старте программа проверяет: запущена она с правами суперпользователя или нет. Если нет, то выводится соответствующее окно, в котором предлагается осуществить запуск «как Root» или продолжить работу с текущими правами. Если вы выберете первое, необходимо будет ввести соответствующий пароль.

Права суперпользователя нужны *Nmap* для осуществления некоторых видов сканирования – например, с целью определения операционной системы на удаленном хосте. Чтобы запустить *Umit* сразу с правами суперпользователя, достаточно дать команду

**\$ sudo umit**

После успешного запуска *Umit* вы увидите главное окно программы (см. рис. 4). Сразу, к сожалению, становится понятно, что *Umit* не имеет русской локализации.

Разберемся с интерфейсом программы. В окне, доступном после запуска, пользователь обнаружит меню и панель инструментов, а под ними, ниже – рабочее поле, где отображаются вкладки.

Вкладка состоит из трех основных частей. В верхней части доступны несколько полей, которые позволяют специфицировать параметры осуществляемого сканирования. Например, здесь доступны такие поля:

► **Target** – поле для указания цели сканирования (и в цифровом формате – IP-адрес, и в символьном – типа `scanme.nmap.org`);

- » Profile – поле позволяет выбрать один из нескольких доступных и ранее подготовленных профилей сканирования (например, Quick Scan – быстрое сканирование);

- » Command – поле, в котором можно редактировать текущую команду сканирования.

Часть окна, расположенная ниже, ориентирована на отображение результатов сканирования и имеет две составляющие. Первая, левая, включает две кнопки: Hosts – отображает просканированные хосты и Service – отображает запущенные на этих хостах и обнаруженные сервисы. Вторая, справа, содержит несколько вкладок:

- » Ports / Hosts – предоставляет информацию об открытых портах и сервисах выбранного хоста;
- » Nmap Output – отображает вывод *Nmap*;
- » Host Details – подробная информация о просканированном хосте;
- » Scan Details – прочие детали сканирования.

Для примера, в поле Target введем следующее:

```
scanme.nmap.org
```

В поле Profile выберем Operating System Detection, после чего значение в поле Command изменится на

```
nmap -O -v scanme.nmap.org
```

После этого нажмем кнопку Scan и запустим сканирование. По завершении сканирования исследуйте вкладки и кнопки нижней части окна *Umit*, о которых мы говорили выше.

Давайте снова осуществим сканирование, с которых мы начали работу с *Nmap* и *Zenmap*. Начнем с пинг-сканирования (здесь и далее – редактируем поле Command, а затем нажимаем кнопку Scan):

```
nmap --sP scanme.nmap.org
```

Вывод будет аналогичен тем, которые мы уже наблюдали при работе с *Nmap* и *Zenmap*.

Теперь воспользуемся типом сканирования, которое позволяло нам при работе с *Nmap* и *Zenmap* определять версии запущенных сервисов:

```
nmap -sV scanme.nmap.org
```

Вывод опять-таки будет аналогичен виденным при работе с *Nmap* и *Zenmap*.

*Umit*, как и *Zenmap*, позволяет сохранять результат сканирования (Scan > Save Scan, или Ctrl+S). Кроме того, *Umit*, как и *Zenmap*, позволяет работать с профилями сканирования (пункт меню Profile), доступен даже специальный пошаговый помощник (Tools > Command Wizard).

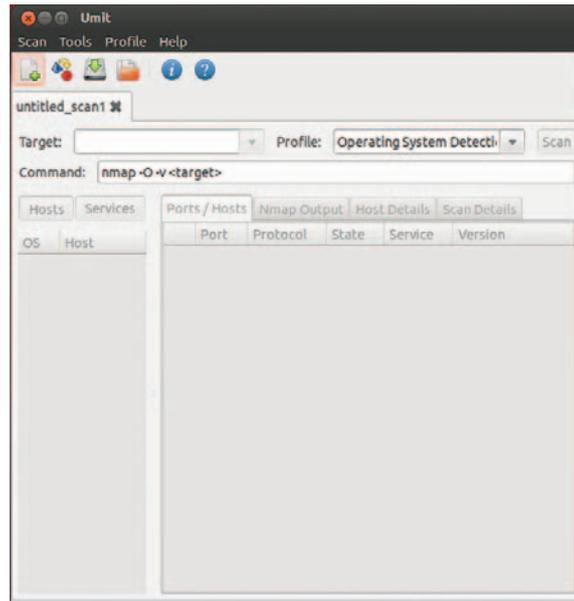
Пожалуй, этого достаточно для успешной работы с *Umit*.

## Что дальше?

Скажем несколько слов о том, где можно получить дополнительную информацию о рассмотренных приложениях. На нашем уроке мы дали общее представление о возможностях рассматриваемых программ и не претендуем на полный охват всех аспектов их использования. Поэтому предлагаем ряд ссылок, которые помогут продолжить изучение.

Для *hping3* доступно руководство, позволяющее начать работу с программой, под названием “Getting started with hping3 [Начинаем работать с hping3]”: <http://wiki.hping.org/94>. Есть и сторонние учебники и статьи о работе с *hping3* – ссылки на них собраны на специальной странице: <http://wiki.hping.org/33>.

Что касается *Zenmap*, то официальный сайт, посвященный *Nmap*, предоставляет специальное Руководство по работе с *Zenmap* (“Zenmap GUI Users’ Guide”) – <http://nmap.org/book/zenmap.html>, которое является составной частью онлайн-версии полезной книги “Nmap Network Scanning [Сетевое сканирование с Nmap]”.



» Рис. 4. Главное окно *Umit*.

На официальном сайте *Umit* документация отсутствует. Правда, если в пункте меню Help выбрать Help, в браузере откроется справка, в которой достаточно подробно рассматривается редактор профилей сканирования (создание профилей, удаление и т.п.). Значительное внимание уделено осуществлению процесса сканирования.

Еще один полезный источник – курс «Инструментальные средства обеспечения безопасности», который доступен на портале Интернет-Университета информационных технологий: <http://www.intuit.ru/departament/security/issec/>. Шестая лекция курса называется «Сканеры портов». В ней дается достаточно подробная вводная информация по использованию программы.

Шестая лекция курса называется «Сканеры портов». В ней дается достаточно подробная вводная информация по использованию программы.

## Заключение

*hping3*, *Zenmap* и *Umit* – инструменты, способные облегчить процесс исследования сети. При этом *hping3* позволяет осуществлять все операции в консоли, а *Zenmap* (официальный графический интерфейс для *Nmap*) и *Umit* – с помощью графического интерфейса. Все три инструмента обеспечения сетевой безопасности доступны из репозитория операционной системы Ubuntu 12.04 (которую мы использовали при работе с ними), что упрощает процесс их установки.

Все три программы обладают гибкими возможностями, а графические интерфейсы для *Nmap* позволяет сделать его использование более удобным для начинающих работать с этим сетевым сканером безопасности. В тоже время, *Umit* уступает по своим возможностям *Zenmap*. Так, например, *Umit* не позволяет отображать топологию просканированной сети. Кроме того, при вводе команды *Zenmap* автоматически определяет цель сканирования и «заполняет» соответствующее поле, а вот *Umit* этого делать не может.

И еще раз напомним, что настоятельно рекомендуется использовать *Nmap* и рассмотренные в статье программы исключительно в исследовательских или рабочих целях, согласовав использование программы со всеми заинтересованными лицами, а также исключительно в рамках действующего законодательства. Дело в том, что, например, многие провайдеры явно запрещают производить сканирование сети, указывая это в соответствующих документах. LXF

## LXC. Малая

Не хватает буквально одного компьютера для нового проекта? Джон Лэйн показывает, как *Контейнеры* Linux снижают загрузку.



Наш эксперт

Набрел на Linux аж в 1994 году, Джон Лэйн на крючке до сих пор. Он не и упомнит, когда в последний раз пользовался Windows.

В главном окне – наш базовый контейнер с деревом процессов и активностью сети. В окне справа показано дерево процессов хоста до и после запуска контейнера.

Мы часто слышим о виртуализации и о преимуществах, которые можно получить, разместив несколько систем на одном компьютере. Обычно это подразумевает запуск разных операционных систем – например, Windows и Linux, и требует наличия программ виртуализации, типа *VirtualBox* или *VMWare*. Программы эти эмулируют устройства и довольно требовательны к системным ресурсам. Однако, если вам нужно запустить несколько рабочих столов Linux на одном компьютере, лучше воспользоваться *Linux Containers (LXC)*: это гораздо более легкое и эффективное решение, и оно является компонентом ядра Linux.

LXC позволяет изолировать сервисы в отдельных контейнерах. Контейнер – нечто вроде виртуальной машины, загружаемой и управляемой независимо от основной системы. Его можно создать и для целой системы, и для одного приложения – например, изолировать браузер или почтовые сервисы от web-сервисов, создав для них отдельные контейнеры.

Контейнерами можно управлять независимо друг от друга: запускать, останавливать, обновлять, переустанавливать, перемещать и удалять. Эта гибкость позволяет упростить работу с большой системой, разбив ее на меньшие, а стало быть, и легче управляемые части.

На нашем уроке мы рассмотрим настройку LXC и создадим базовый контейнер, который можно будет загрузить и войти в него. Затем мы расширим его – пусть поддерживает графический рабочий стол. LXC имеется в репозиториях большинства дистрибутивов. Ubuntu в своем последнем релизе – 12.04 Precise Pangolin – продвинулась еще на шаг вперед и теперь полностью поддерживает LXC «прямо из коробки». Тем не менее, чтобы воспользоваться всеми возможностями LXC, нам обязательно нужно понять, как они работают.

LXC, с помощью контрольных групп ядра Linux, сужает возможности дерева процессов, выставляя ограничения атрибутов процесса, таких как потребление процессора и памяти. При запуске контейнера создается контрольная группа на основе настроек контейнера, затем управление передается процессу *init*

корневой файловой системы контейнера. Процесс *init* выполняет обычную загрузку системы, которая завершается появлением строки входа в систему.

### Настройка хоста

Итак, с чего начнем? LXC состоит из двух частей – компонентов ядра и пользовательских утилит. Поэтому убедитесь, что у вас подходящее ядро, версии не менее 2.6.27 (командой `uname -r`), и установите пользовательские утилиты из менеджера пакетов дистрибутива. В Ubuntu можно выполнить команду `sudo apt-get install lxc`. В Arch Linux они находятся в пользовательском репозитории Arch (Arch User Repository), поэтому нужно скомпилировать и собрать их там.

Установив утилиты, проверьте, что необходимые компоненты ядра включены:

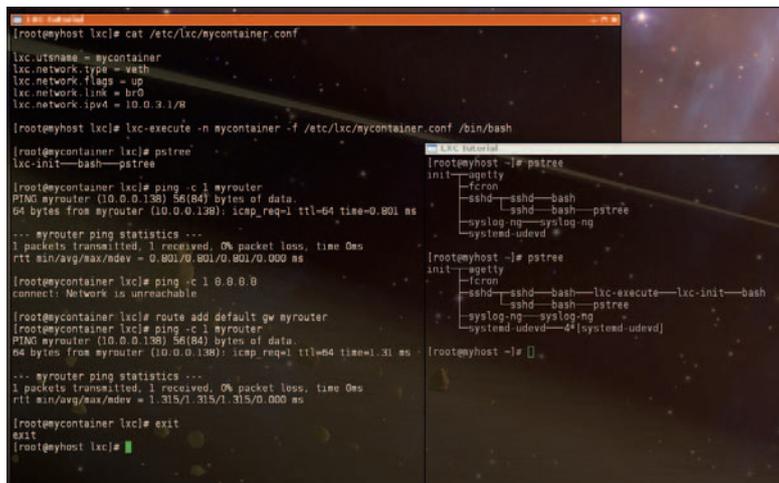
```
# lxc-checkconfig
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled
--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled
--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
$File capabilities: enabled
```

Если параметр **Cgroup namespace** имеет значение **required** вместо **enabled**, это известная ошибка, которую можно проигнорировать. Если есть другие отличия, перекомпилируйте ядро, включив эти опции. В большинстве дистрибутивов они включены по умолчанию, и проблем тут не должно быть.

Затем убедитесь, что файловая система ядра **cgroup** смонтирована. Делается это командой

```
# mount | grep cgroup
Если она не смонтирована, смонтируйте ее. Ее обычная точка монтирования – /cgroup:
# mkdir /cgroup
# echo "none /cgroup cgroup defaults 0 0" >> /etc/fstab
# mount /cgroup
```

Следующее, что нам понадобится – сетевой стек, который поддерживает хост и все контейнеры. Есть несколько способов его реализации, но проще всего применить мост. В Ubuntu 12.04 при установке LXC мост создается автоматически. В других ди-



# ВИРТУАЛИЗАЦИЯ

стрибутивах установите и настройте **bridgeutils** (советы найдутся в наших архивах – [www.linuxformat.com/archives](http://www.linuxformat.com/archives)). Итак, пусть наш мост создан и называется **br0**.

## Наш первый контейнер

Попробуем запустить наш первый контейнер, просто для проверки, что все работает:

```
# lxc-execute -n mycontainer /bin/bash
# pstree
lxc-init---bash---pstree
# exit
```

Может показаться, что команда **lxc-execute** ничего не сделала; но она запустила *Bash* внутри контейнера и оставила вас в командной строке *Bash* этого контейнера. Взгляните на дерево процессов – в контейнере их всего три. Наш контейнер весьма примитивен, и он по-прежнему пользуется общей корневой файловой системой и сетью. Но с несложным файлом конфигурации это можно изменить. Создайте новый файл с именем **/etc/lxc/mycontainer.conf**:

```
lxc.utsname = mycontainer
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.ipv4 = 10.0.3.1/8
```

Он дает нашему контейнеру собственное имя (его имя хоста – **mycontainer**) и сетевой адрес (выберите подходящий свободный IP-адрес). Тип сети **veth** велит *LXC* использовать сетевой мост, а сетевая ссылка – это имя моста, в данном случае **br0**. Запустите контейнер и загляните внутрь:

```
# lxc-execute -n mycontainer -f /etc/lxc/mycontainer.conf
/bin/bash
# hostname
mycontainer
# ping myhost
```

У вас есть доступ к компьютерам локальной сети, но не более того. Чтобы выйти за пределы локальной сети, понадобится создать маршрут к шлюзу по умолчанию – например, так: **route add default gw myrouter**. По окончании работы с контейнером выйдите из него командой **exit**.

## Загружаем его

Наш пример рассматривал контейнер приложений – в нем в контейнере запускалось одно приложение (*Bash*). Уже замечательно; но по-настоящему-то полезны системные контейнеры, в которых своя корневая файловая система и которые выглядят как отдельные компьютеры.

Продемонстрируем это: создадим системный контейнер. Также мы добавим к нему графический рабочий стол, но пока в нем будет базовая загружаемая система. Изменим файл **/etc/lxc/mycontainer.conf**, чтобы дать контейнеру доступ к необходимым ресурсам:

```
lxc.utsname = mycontainer
lxc.network.type = veth
```

```
lxc.network.flags = up
lxc.network.link = br0
lxc.network.name = eth0
# only explicit device access
lxc.cgroup.devices.deny = a
# Memory Devices
lxc.cgroup.devices.allow = c 1:3 rwm # /dev/null null stream
lxc.cgroup.devices.allow = c 1:5 rwm # /dev/zero zero stream
lxc.cgroup.devices.allow = c 1:7 rwm # /dev/full full stream
lxc.cgroup.devices.allow = c 1:8 rwm # /dev/urandom blocking
random stream
lxc.cgroup.devices.allow = c 1:9 rwm # /dev/random non blocking
stream
# Terminals
lxc.tty = 2 # local ttys: /dev/tty1 and /dev/tty2
lxc.cgroup.devices.allow = c 4:0 rwm # /dev/tty0 current virtual
terminal
lxc.cgroup.devices.allow = c 4:3 rwm # /dev/tty3 host's /dev/tty3
lxc.cgroup.devices.allow = c 5:0 rwm # /dev/tty current tty device
lxc.cgroup.devices.allow = c 5:1 rwm # /dev/console system
console
lxc.cgroup.devices.allow = c 5:2 rwm # /dev/ptmx pseudo terminal
creator
lxc.cgroup.devices.allow = c 136:* rwm # /dev/pts/* psuedo
terminal slaves
# root filesystem
lxc.rootfs = /srv/lxc/mycontainer
# mounts
lxc.mount.entry = noPress ALT-F3 to switch to tty3 or sne /n
LXCsrv/lxc/mycontainer/dev/pts devpts defaults 0 0
lxc.mount.entry = none /srv/lxc/mycontainer/dev/shm tmpfs
defaults 0 0
lxc.mount.entry = none /srv/lxc/mycontainer/proc proc defaults 0 0
lxc.mount.entry = none /srv/lxc/mycontainer/sys sysfs defaults 0 0
```

Теперь в файле конфигурации гораздо больше параметров. Настройки сети похожи на предыдущие, только назначение IP-адреса предоставлено процедуре загрузки контейнера.

Скорая помощь

Примеры конфигурации можно найти в **/usr/share/doc/lxc/examples**.

Скорая помощь

Вам помогут в IRC-канале **#lxccontainers** или в рассылке **lxc-users**. См. страницу поддержки *LXC* – <http://sourceforge.net/projects/lxc/support>.

## OpenVZ

Аналогичная технология – *OpenVZ*, и ее код составляет большую часть кода *LXC*. Технология *OpenVZ* является более зрелой, но для ее работы нужно залатать ядро, и ее поддерживают немногие дистрибутивы (в Ubuntu поддержка прекратилась после 8.04 Hardy Heron). Документация *OpenVZ* основана на Fedora Core, RHEL и CentOS.

Если вы хотите подобраться поближе к «уровню предприятия» и у вас дистри-

бутив на базе Red Hat, зайдите на [www.openvz.org](http://www.openvz.org). Однако *LXC* реализованы в основной версии ядра и находятся в активной разработке, поэтому мы уделяем им внимание.

Знать об *OpenVZ* полезно еще и потому, что здесь есть готовые шаблоны для различных дистрибутивов, и ими можно воспользоваться в *LXC*. Ищите шаблоны по ссылке <http://wiki.openvz.org/Download/template/precreated>.

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

## Скорая помощь



Для создания резервной копии контейнера скопируйте его корневую файловую систему. Лучше делать это при остановленном контейнере. Резервная копия поможет вернуть контейнер в предыдущее состояние.

## Скорая помощь



Тип терминала в контейнере проверить легко. Чтобы вывести список терминалов, выполните команду `ls -l /dev/tty?` в контейнере. Взгляните на самое большое число: для терминалов основного компьютера это 4, для локальных терминалов – 136.

Затем нам нужно ограничить контейнер в правах, иначе он распышется и будет покушаться на устройства главного компьютера. Лучше всего запретить все подряд, разрешив только самое необходимое для конкретной конфигурации. В последующих разделах мы разрешим доступ к повсеместно используемым устройствам памяти, предполагаемых в типичной системе.

## Три — волшебное число

Внутри контейнеров можно пользоваться терминальными устройствами трех типов. Если процессу необходим терминал, например, при подключении к контейнеру по SSH, то по его запросу создается псевдотерминал. Никаких специальных настроек, кроме настройки доступа в `/dev/ptmx` и `/dev/pts/*` аналогично нашему примеру, не требуется.

Два других типа – различные представления базового устройства `tty`. У правильно настроенного контейнера есть доступ к устройствам `tty` главного компьютера, а также к собственным локально созданным устройствам `tty`. Параметр `lxc.tty` определяет, сколько устройств можно создать локально внутри контейнера. Например, если он равен 2, то в LXC автоматически появятся локальные терминалы `/dev/tty1` и `/dev/tty2`, доступные по команде `lxc-console`. Чтобы воспользоваться терминалом основного компьютера, выберите устройство с большим номером и настройте доступ к нему (в нашем примере – `/dev/tty3`). Контейнеру можно пользоваться терминалом главного компьютера, когда главному компьютеру терминал не нужен: сделайте так, чтобы процессы `getty` не подключались к этому `tty` (в большинстве дистрибутивов это настраивается в `/etc/inittab`).

В нашем примере в контейнере взят терминал основного компьютера `/dev/tty3`, доступный как виртуальный терминал (для его открытия нажмите `Alt+F3`). В контейнере также есть терминалы `/dev/tty1` и `/dev/tty2` – доступа к обоим обеспечит команда `lxc-console`. Последними в нашей конфигурации задаются точки монтирования файловой системы контейнера; главная из них – корневая файловая система контейнера. Это обычный каталог на главном компьютере. Выберите подходящий каталог – скажем, `/srv/lxc/mycontainer` – и заполните его с помощью соответствующих утилит вашего дистрибутива.

В Arch Linux вам поможет утилита `mkarchroot` (предварительно установите пакеты `devtools` и `base-devel`):

```
# mkdir /srv/lxc
# mkarchroot /srv/lxc/mycontainer base
```

В Debian/Ubuntu для этого служит утилита `debootstrap`:

```
debootstrap --arch=i386 precise /srv/lxc/ubuntu_debootstrap
http://archive.ubuntu.com/ubuntu
```

Чтобы подготовить корневую файловую систему контейнера для LXC, в ней нужно кое-что поменять. При запуске системного контейнера LXC вызывает процесс `init` внутри контейнера, запускающий загрузочную последовательность. Но чтобы загрузка

работала внутри контейнера, ее стоит подправить, так как многое из обычного запуска и выключения системы в контейнере не требуется. Меняют для этого скрипты `init`, а конкретные действия записывают от дистрибутива.

В Ubuntu версий менее 12.04 есть пакет `lxcguest`, содержащий необходимые поправки. В версии 12.04 стандартные настройки учли настройки LXC, и никаких поправок не требуется. В наших архивах есть примеры скриптов запуска для Arch Linux. Затем нужно создать файлы для тех устройств, доступ к которым был разрешен в файле конфигурации. Обычно деревом устройств в `/dev` управляет демон `udev`, но в контейнере это не работает. Мы должны вручную создать файлы устройств в соответствии с файлом конфигурации. Для начала создадим общие устройства, необходимые большинству контейнеров:

```
cd /srv/lxc/mycontainer/dev
mknod -m 666 null c 1 3 # null device
mknod -m 666 zero c 1 5 # zero device
mknod -m 666 full c 1 7 # full device
mknod -m 666 random c 1 8 # blocking random device
mknod -m 666 urandom c 1 9 # non blocking random device
mknod -m 666 tty0 c 4 0 # system console
mknod -m 666 tty c 5 0 # process's current tty device
mknod -m 600 console c 5 1 # system console aka tty0
mknod -m 666 ptmx c 5 2 # pseudo terminal creator
mknod -m 600 initctl p # init control channel
mkdir -m 755 pts # pseudo terminal slaves
mkdir -m 1777 shm # shared memory
ln -s /proc/self/fd fd # for backward compatibility
```

Также нужно создать устройства `tty`. В нашей конфигурации их имеется три:

```
mknod -m 666 tty1 c 4 1 # tty1
mknod -m 666 tty2 c 4 2 # tty2
mknod -m 666 tty3 c 4 3 # tty3 (host vt)
```

Перед запуском контейнера войдем в него через `chroot` и выполним все необходимые действия для настройки. Например, можно задать пароль пользователя `root`, создать пользователей и установить программы. После монтирования нескольких независимых файловых систем `chroot` получит функциональность, достаточную для установки пакетов программ:

```
# mount proc -t proc /srv/lxc/mycontainer/proc
# mount devpts -t devpts /srv/lxc/mycontainer/dev/pts
# chroot /srv/lxc/mycontainer
```

Задайте пароль пользователя `root` и создайте учетные записи других пользователей:

```
echo "root:myrootpassword" | chpasswd
useradd -c 'My User' -m -s /bin/bash myuser
echo "myuser:mypassword" | chpasswd
```

На этом настройка контейнера закончена – остальное можно сделать после его загрузки. Так что пока выйдем из `chroot`:

## Что такое контрольные группы?

В примерах конфигурации LXC вы видели `cgroup`. Это контрольные группы, которые появились в ядре Linux с версии 2.6.24. Они позволяют выделить группы процессов и управлять выдачей им ресурсов, таких как процессор, память и ширина канала сети. Именно контрольные группы делают виртуализацию контейнеров возможной, а LXC выполняют всю трудную работу, освобождая от нее вас. Пара-

метры `cgroup`, указываемые в файле конфигурации контейнера, разнообразны. Для просмотра доступных параметров запустите контейнер и выполните команду `ls /cgroup/mycontainer` с главного компьютера. Вот несколько полезных параметров:

- » `cpuset.cpus` ограничивает процессоры, доступные контейнеру (например, 0-1 – процессоры 0 и 1).
- » `cpuset.shares` задает большее или меньшее коли-

чество циклов процессора. По умолчанию используется 1024, это нормальное значение; 512 – вдвое меньше. `memory.max_limit_in_bytes` задает максимум выделяемой памяти (например, 512M, 1G и т.д.).

К сожалению, этих настроек много для нашей статьи. Узнать о них подробнее можно в документации ядра: <http://kernel.org/doc/Documentation/cgroups>.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

```
# exit
# umount /srv/lxc/mycontainer/dev/pts
# umount /srv/lxc/mycontainer/proc
```

Контейнер настроен; пора его запустить. Создайте его в *LXC* и выполните команды:

```
# lxc-create -n mycontainer -f /etc/lxc/mycontainer.conf
'mycontainer' created.
```

```
# lxc-start -n mycontainer
```

Если все хорошо, контейнер загрузится. Нажмите **Alt+F3**, чтобы переключиться на *tty3*, или откройте консоль *LXC* для входа в систему:

```
# lxc-console -n mycontainer
```

Зайдите в контейнер от имени *root* с паролем, заданным ранее. Осмотрите новую систему и выполняйте всякие команды. Закончив, отключитесь от консоли, нажав **Ctrl+A**, затем **q**. Если перед отключением не выйдете из системы, позже можно подключиться к тому же сеансу.

Вернувшись на главный компьютер, рассмотрим некоторые утилиты *LXC*. Команда **lxc-ls** выводит список всех известных контейнеров в одной строке и всех запущенных в следующей. Дерево процессов запущенного контейнера покажет команда **lxc-ps -n mycontainer**. Это обертка вокруг стандартной команды **ps**, поэтому вслед за именем контейнера можно указывать любые аргументы **ps**. Процессы контейнера также прекрасно показывает утилита **pstree**. Остановить контейнер можно командой **lxc-stop -n mycontainer**, но это «жесткая» остановка – все равно что выдернуть вилку из розетки. Лучше выполнить внутри контейнера команду **shutdown -h now**. Теперь проверим-ка наш **ssh**:

```
# ssh mycontainer
# shutdown -h now
```

## Рабочий стол внутри контейнера

Пора сделать с нашим контейнером кое-что более полезное. Мы установим базовую версию рабочего стола X и свяжем его с одним из виртуальных терминалов основного компьютера. Это базовая версия *Xorg* с драйверами *VESA* – джентльменский набор для запуска рабочего стола внутри контейнера. Употребим нашу мышь и клавиатуру и свяжем рабочий стол с виртуальным терминалом 9 (**Alt+F9**). Добавьте следующие параметры в файл конфигурации контейнера **/etc/lxc/mycontainer.conf**:

```
# VESA XOrg Desktop
lxc.cgroup.devices.allow = c 4:9 rwm # /dev/tty9 X Desktop
lxc.cgroup.devices.allow = c 13:* rwm # /dev/input/* input devices
lxc.cgroup.devices.allow = c 10:63 rwm # /dev/vga_arbiter
lxc.cgroup.devices.allow = c 1:1 rwm # /dev/mem
```

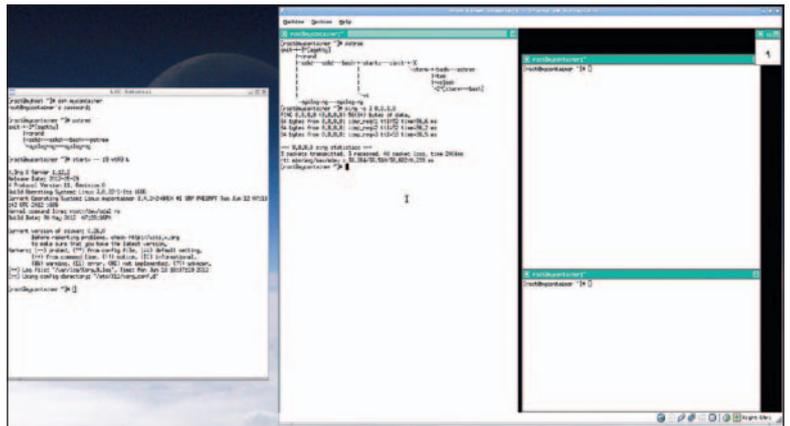
Указанные права дают доступ к необходимым устройствам ввода и вывода. При настройке другого графического драйвера потребуется доступ к другим устройствам. Создайте соответствующие файлы устройств:

```
cd /srv/lxc/mycontainer/dev
mkdir -m 755 input # input devices
mknod -m 666 input/mice c 13 63 # mice
mknod -m 666 tty9 c 4 9 # vt for display
mknod -m 666 vga_arbiter c 10 63 # vga display
mknod -m 666 mem c 1 1 # memory
```

При любых изменениях в файле конфигурации *LXC* должен удалить и снова создать контейнер. При этом никакие файлы не удаляются. Убедившись, что контейнер остановлен, воссоздайте его, перезапустите и войдите в контейнер:

```
lxc-destroy -n mycontainer
lxc-create -n mycontainer -f /etc/lxc/mycontainer.conf
screen -dmS mycontainer lxc-start -n mycontainer
ssh root@mycontainer
```

Внутри контейнера установите необходимые пакеты *Xorg* для рабочего стола. В Arch Linux:



```
pacman -S xorg-server xorg-xinit xf86-input-{keyboard,mouse}
xf86-video-vesa
```

```
pacman -S xorg-twm xorg-xclock xterm
```

Единственная настройка *Xorg* нужна для клавиатуры и мыши, потому что для автоматического определения устройств в *Xorg* используется **udev**, а в контейнере его нет. Настройки *Xorg* хранятся в новом файле **/etc/X11/xorg.conf.d/10-input.conf**:

```
# /etc/X11/xorg.conf.d/10-input.conf
```

```
Section "ServerFlags"
```

```
Option "AutoAddDevices" "False"
```

```
EndSection
```

```
Section "ServerLayout"
```

```
Identifier "Desktop"
```

```
InputDevice "Mouse0" "CorePointer"
```

```
InputDevice "Keyboard0" "CoreKeyboard"
```

```
EndSection
```

```
Section "InputDevice"
```

```
Identifier "Keyboard0"
```

```
Driver "kbd"
```

```
Option "XkbLayout" "gb"
```

```
EndSection
```

```
Section "InputDevice"
```

```
Identifier "Mouse0"
```

```
Driver "mouse"
```

```
Option "Protocol" "auto"
```

```
Option "Device" "/dev/input/mice"
```

```
EndSection
```

Наконец, теперь можно запустить внутри контейнера *Xorg* и увидеть рабочий стол:

```
startx -- :9 vt09
```

Только что мы создали контейнер с доступом к устройствам главного компьютера. Мы вручную создали необходимые файлы устройств и установили нужные пакеты *Xorg*. И мы запустили рабочий стол внутри контейнера. А если у вас последняя версия Ubuntu 12.04 Precise Pangolin, вы можете воспользоваться шаблонами Ubuntu Cloud. Создать новый контейнер просто:

```
$ sudo apt-get install lxc
$ sudo lxc-create -t ubuntu-cloud -n mycontainer
$ sudo lxc-start -n mycontainer
```

Желая воспользоваться сервисами главного компьютера (как в нашем примере с запуском рабочего стола), как и прежде, нужно изменить настройки *LXC*. Образы облака можно загрузить и вручную с <https://cloud-images.ubuntu.com>, но файл настроек *LXC* придется создать самостоятельно. Примеры – на нашей странице с архивами для этого номера ([www.linuxformat.com/archives](http://www.linuxformat.com/archives)).

*LXC* – относительно новый компонент платформы Linux. Это прекрасный проект, и Canonical использует его даже в *Juju*. А поскольку *LXC* получили прописку в основной версии ядра, их ждет большое будущее. **LXF**

► Простой, но подлинно рабочий стол внутри контейнера. Слева – оболочка, его запустившая. Проверьте вывод **pstree** до и после запуска рабочего стола.

Скорая помощь

Если вы запускаете контейнер с помощью **screen**, при необходимости его можно отключить от консоли (нажмите **Ctrl+a**, затем **d**).

# ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу [answers@linuxformat.ru](mailto:answers@linuxformat.ru), и мы найдем ответ.

В этом месяце мы ответим на вопросы про...

- 1 Конвертацию видеороликов в формат DVD
- 2 Проблемы с тачпадом
- 3 Беспроводные сети в Ubuntu
- 4 Поддержку устройств в Linux
- 5 Недомогание Asus Eee PC
- 6 Прокси-сервер для старого компьютера

## 1 Видео в DVD

В Недавно я предложил другу записать на DVD видео с его праздника. Съемка шла на видеокамеру Papasonic со смарт-картами объемом 8 и 16 Гб, каждый клип представлен четырьмя файлами: `.cont`, `.rmpd`, `.mp4` и `.tmb`. *K3b* выдала сообщение о том, что данных недостаточно, *Brasero* открыл несколько файлов и завис. Боюсь, с *Brasero* у меня всегда такая история. *DeVeDe* вроде работает, но добавлять клипы перетаскиванием можно только по одному — а их более тысячи. В отчаянии я добавил две сотни клипов в *OpenShot*, который сразу «упал» без видимой причины.

Наконец, я сдался, открыл свой старый раздел с XP и запустил *Nero Vision* десятилетней давности. Это было здорово: нажав клавишу Shift и выделяя файлы мышью, я смог добавить сразу 98 роликов и еще одним щелчком мыши объединить их. Затем программа сконвертировала все файлы и записала их на DVD. Почему в Linux такого нет? Какой программой я должен был воспользоваться? В командной строке мне пришлось бы набирать идентификатор каждого ролика, но это же нереально.

Джон Тейлор [John Taylor]

Очень похоже, что клипы уже загружены в *Photo Fun Studio*, программу для редактирования клипов в Windows, предоставляемую с камерой. Она создает для каждого ролика дополнительные файлы, а Вам нужен только сам ролик. Скорее всего, у него будет самая ранняя дата создания, и уж точно он будет самым увесистым. Обычно это файл MTS, но, кажется, камера была переведена в режим записи MPEG, и Вам нужны файлы MP4. Если есть сомнения, попробуйте загрузить файлы в видеопроигрыватель, типа *MPlayer* или *VLC*, чтобы понять, в каком из них содержится видеоролик.

Скорее всего, программы «падают» именно при загрузке посторонних файлов. Вы также превышаете лимит формата DVD в 99 названий на диск. Если оставить только файлы MP4 или MTS и не добавлять на один диск слишком

много, *DeVeDe* работает прекрасно. Чтобы не загружать ролики по одному, откройте файловый менеджер, выберите видеофайлы и перетащите их на панель Files [Файлы] *DeVeDe*. Выбрать несколько файлов в командной строке очень просто благодаря маске имени. Например, `*.mp4` представляет собой список всех файлов с расширением `.mp4` в алфавитном порядке перед передачей нужной программе.

Создание DVD из набора файлов разных форматов включает несколько этапов; но программы вроде *DeVeDe* их от Вас прячут. В командной строке похожую задачу выполняет утилита *tovid* ([http://tovid.wikia.com/wiki/Tovid\\_Wiki](http://tovid.wikia.com/wiki/Tovid_Wiki)). Вот команда, конвертирующая все файлы MP4 в текущем каталоге, если их не более 98 (одно из 99 доступных названий — само меню DVD):

```
tovid disc -files *.mp4 -menu-title "My video"
-textmenu -out mydvd
tovid dvd mydvd
```

Первая команда выполняет конвертацию и создает структуру файлов DVD, вторая прожигает все это на DVD.

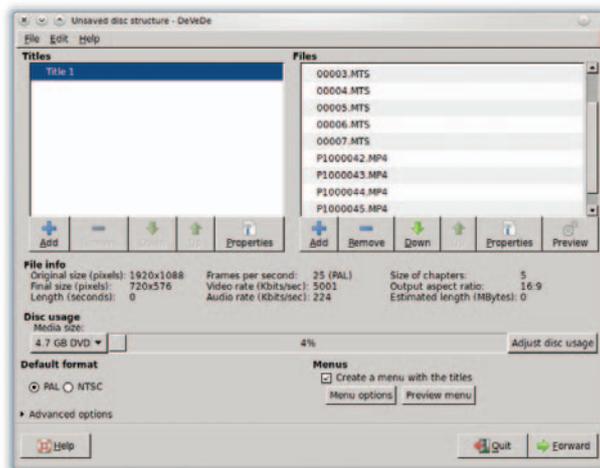
## 2 Eee PC-недотрога

В Мне нужен ваш совет о том, как включать и выключать сенсорную площадку-тачпад на Eee PC 1011PX, что я хотел бы делать клавишами Fn+F3. Сейчас это не работает.

Я пользуюсь *Xubuntu 11.10* и установил *Jupiter 0.1.2*, программу для управления тачпадом. Но если выключить тачпад и не включить его снова перед выключением нетбука, то потом он не включается вообще. Остальные комбинации Fn+клавиша работают нормально.

Дуглас Макдоналд [Douglas Macdonald]

О В состав *Jupiter* входит несколько скриптов, автоматизирующих некоторые из его функций. Они живут в каталоге `/usr/lib/jupiter/scripts`, и тот, что Вам нужен, называется *touchpad*. Если запустить его с аргументом `restore`, будет восстановлено сохраненное состояние, а если без аргумента — тачпад будет включен или выключен. Ваш вывод `xev` показывает, что комбинация Fn+F3 возвращает код `XF86TouchpadToggle`, который и должна возвращать. Следовательно, все настроено правильно. Нужно просто прика-



DeVeDe создает DVD из видеороликов, но не превышайте пределов, установленных спецификацией DVD.

зать *Xfce* связать скрипт тачпада с клавишей тачпада. Для этого зайдите в Settings > Settings Editor [Настройки > Редактор настроек] и выберите параметр `xfce4-keyboard-shortcuts`.

Но и тут все-таки нужно не забывать включать тачпад перед выключением компьютера. *Jupiter* сохраняет текущее состояние тачпада в файле и восстанавливает его при загрузке системы (или при запуске команды `/usr/lib/jupiter/scripts/touchpad restore`). Чем и можно воспользоваться, записывая в этот файл единицу с помощью скрипта перед выключением или при запуске системы. Для этого откройте файл `/etc/rc.local` под пользователем `root` и добавьте все необходимые команды перед завершающей `exit 0` — получится нечто вроде

```
echo 1 >/var/jupiter/touchpad_saved
exit 0
```

Команды из `/etc/rc.local` выполняются при загрузке системы, но если Вы хотите запустить свою именно при выключении системы, потребуется создать скрипт в `/etc/rc6.d`, и его имя должно начинаться с `K99`. Добавить в него нужно то же самое — кроме того, Вам понадобится строка из верхней части скрипта `rc.local`, и весь файл будет выглядеть так:

```
#!/bin/sh -e
echo 1 >/var/jupiter/touchpad_saved
exit 0
```

Затем сделайте файл исполняемым:

```
sudo chmod +x /etc/rc6.d/K99touchpad
```

Вывод команды `xinput`, который Вы привели в письме, говорит, что нужно было применить немного другой подход и управлять тачпадом напрямую с `xinput` (скрипты *Jupiter* используют `xinput`), хотя это могло бы означать, что тачпад

не обязательно находится в том состоянии, которого ожидает *Jupiter*. Получите идентификатор тачпада командой

```
xinput list
```

Затем снова воспользуйтесь **xinput**, чтобы определить код отключения устройства:

```
xinput list-props <идентификатор>
```

Среди свойств должно быть такое:

```
Device Enabled (135): 1
```

Число в скобках – код устройства, 1 – текущее состояние (0 – отключено). Поэтому включить тачпад можно командой

```
xinput set-prop <id-number> 135 1
```

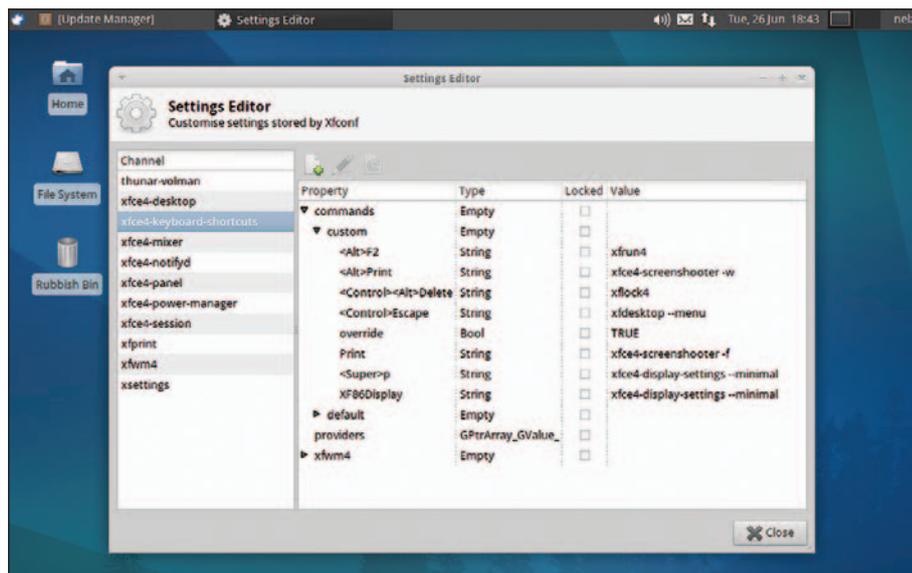
Добавьте эту команду в `/etc/rc.local` или в `/etc/rc6.d`.

### 3 Шаткая беспроводная сеть

В Я установил Ubuntu 12.04 LTS с Windows 7 на нетбук Toshiba Netbook NB510-119. Win 7 нормально подключается к Интернету по беспроводной сети роутера Thomson Speedtouch ST585 v6. Ubuntu подключается при загрузке, но через несколько минут отключается. Страница настройки роутера открывается и доступна в течение нескольких минут, но потом я получаю сообщение об ошибке загрузки страницы в *Firefox*.

Билл Причард [Bill Pritchard]

О Другие сообщения об этой ошибке говорят о том, что она связана с качеством сигнала. Когда оно падает ниже определенного уровня, драйвер, видимо, входит в шторм, пытаясь добиться лучшего соединения. Чтобы это проверить, положите нетбук рядом с роутером, чтобы уровень сигнала был максимальным. Также может помочь переключение на другой канал беспроводной связи – если тем же каналом пользуются соседи, это снижает качество сигнала.



➤ Для добавления или изменения комбинаций клавиш в Xfce зайдите в Редактор настроек [Settings Editor].

ла. Если Вы пользуетесь каналом по умолчанию, это с большой вероятностью поможет.

Но лучше будет установить новые драйверы с сайта [www.realtek.com/downloads/downloadsView.aspx?Langid=1&PNid=21&PFid=48&Level=5&Conn=4&DownTypeID=3&GetDown=false&Downloads=true](http://www.realtek.com/downloads/downloadsView.aspx?Langid=1&PNid=21&PFid=48&Level=5&Conn=4&DownTypeID=3&GetDown=false&Downloads=true).

Найдите на этой странице драйверы для RTL8188CE и загрузите драйвер Linux для ядра 2.6.24 (и более поздних версий, до 3.2.x). Это tar-архив. Откройте терминал, зайдите в каталог, куда загружен файл, распакуйте архив и войдите в каталог:

```
tar xf 92ce_se_de_linux_mac80211_0005.1230.2011.tar.gz
```

```
cd rtl_92ce_92se_92de_linux_mac80211_0005.1230.2011
```

Драйвер нужно скомпилировать под Ваше ядро – это совсем не так страшно, как кажется, но нужно убедиться, что установлен пакет **build-essential**. Он содержит программы, необходимые для компиляции и установки программ. Вернитесь в терминал и скомандуйте:

```
sudo make
sudo make install
```

Первая команда компилирует программу, проверяет отсутствие ошибок компиляции и запускает вторую команду, которая устанавливает драйвер и прошивку туда, где ядро сможет найти их. После перезагрузки будет использоваться новый драйвер.

Так как драйвер компилируется и устанавливается под конкретное ядро, после установки любых обновлений ядра этот процесс придется повторить (*UpdateManager* может предложить установить новое ядро с исправлениями ошибок или улучшением безопасности), поэтому архив не удаляйте. Для удаления драйвера повторите процесс, заменив последнюю команду следующей:

```
sudo make uninstall
```

### 4 Поддержка устройств

В Я всерьез задумался о переходе с Windows на Linux и хотел бы узнать, насколько легко устанавливать принтеры и другие устройства. В Windows я привык к идее “Plug and play” и хотел бы узнать, есть ли какие-то отличия в Linux. Дистрибутив выбираю либо Linux Mint, либо Ubuntu 12.04.

Майк Холлидей [Mark Halliday]

О Поддержка устройств в Linux устроена совсем иначе по сравнению с Windows и способна немного сбить с толку новых пользователей. ОС Windows поддерживает очень немногие устройства, полагаясь на то, что производители устройств предоставят драйверы



## Коротко про...

# Алиасы оболочки

Если вы постоянно запускаете команду с одними и теми же аргументами, не проще ли задать их по умолчанию или приписать несколько команд для разных вариантов запуска программы с разными аргументами? Все это и многое другое возможно с псевдонимами (алиасами) оболочки.

Скорее всего, у вас уже есть несколько алиасов. Наберите `ll`, и вы увидите список файлов с подробной информацией о них – размерами файлов, датами и владельцами. Это возможно благодаря тому, что во многих дистрибутивах `ll` – это алиас для `ls -l`. Для просмотра существующих алиасов выполните команду `alias` без аргументов. Чтобы создать алиас, выполните команду:

```
alias la="ls -lhA --color=auto"
```

Теперь, набрав `la` (`list all` – вывести список всех файлов) в текущем сеансе оболочки, вы увидите список файлов с подцветкой и подробной информацией о файлах, включая скрытые. Команда

вместо алиаса подставляется перед выполнением его оболочкой. Но если открыть новую оболочку, алиаса там не будет – команду `alias` придется выполнять для каждой новой оболочки. Этот процесс можно автоматизировать, потому что перед запуском оболочки всегда анализируются несколько файлов. Системные настройки хранятся в `/etc/profile`; алиасы по умолчанию обычно записываются сюда. Кроме того, в файле настройки оболочки можно задавать собственные алиасы. Для Bash это `~/.bashrc` или `~/.bash_profile`. Первый используется в интерактивной оболочке, вроде *Konsole* или *xterm*. Второй – в оболочке входа в систему. Обычно команды-алиасы добавляются в `~/.bashrc`. Алиас может выполнять более одной команды, например:

```
alias foobar="foo --foo-opts; bar --bar-opts"
```

поэтому простые наборы команд можно сворачивать в алиасы, а не в скрипты.

на диске или в Интернете, и новички в Linux по аналогии начинают искать драйверы для Linux на сайтах производителей.

Linux в основном поддерживает устройства изнутри: сетевые адаптеры, устройства ввода и видеокарты в большинстве случаев поддерживаются напрямую ядром. Это означает, что драйверы для большинства устройств уже установлены – и будут загружены автоматически, как только система обнаружит устройство.

Принтеры и сканеры работают не под управлением ядра, а под управлением систем CUPS и SANE соответственно. Существует серверное ПО, с помощью которого с принтерами и сканерами могут работать другие программы, оно содержит драйверы для большинства известных устройств. Иногда для установки драйверов конкретного принтера или сканера нужно установить отдельный пакет с драйверами из менеджера пакетов дистрибутива. Например, принтеры и сканеры HP поддерживаются в Linux открытыми драйверами, собранными в отдельный пакет **hplip** (HP Linux Imaging and Printing – сканирование и печать HP в Linux). Вам не придется искать драйверы на сайтах производителей – все должно быть доступно в менеджере пакетов дистрибутива.

Еще одна область, которая раньше была проблемной – беспроводная сеть, но сейчас ситуация улучшилась: большинство чипсетов поддерживаются либо напрямую ядром Linux, либо дополнительными пакетами с драйверами, и в большинстве дистрибутивов устройства хорошо определяются и для них загружаются соответствующие драйверы. В некоторых из этих драйверов, например, в драйверах для видеокарт Nvidia, используется проприетарный код, и Вам может понадобиться включить соответствующую опцию в менеджере пакетов.

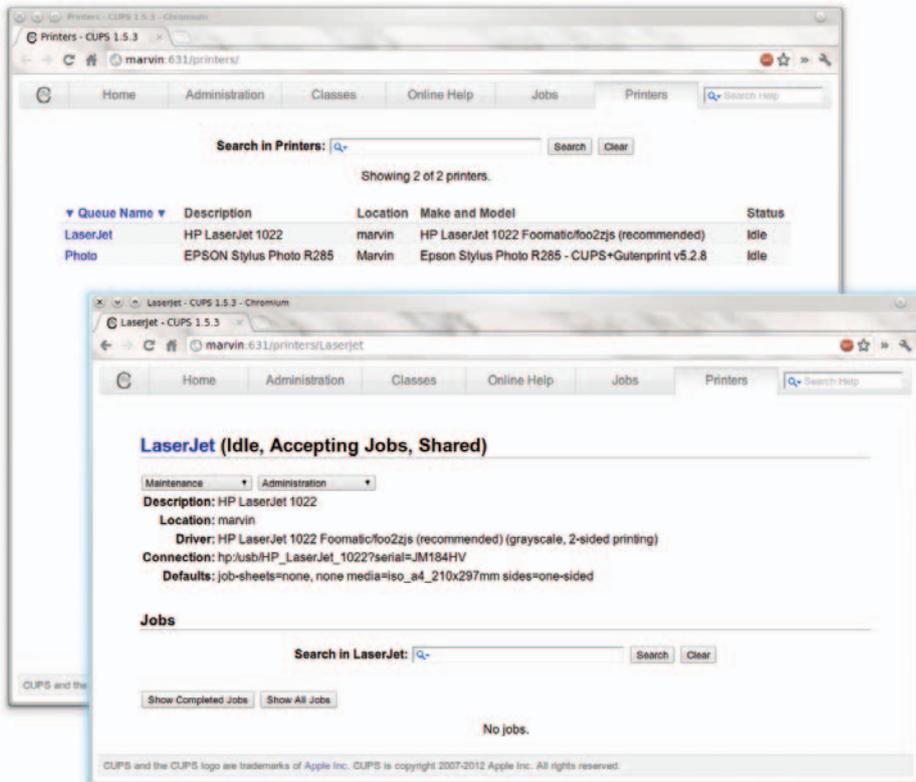
Некоторые устройства не слишком хорошо работают с Linux, потому что производители не предоставляют ни драйверов, ни документации для их разработки, но их число уменьшается.

На сайтах некоторых дистрибутивов можно найти список поддерживаемых устройств, но есть и несколько специальных сайтов, на которые стоит зайти, прежде чем покупать устройства, с которыми Вы планируете работать в Linux.

На <http://linuxprinting.org> есть исчерпывающая база данных принтеров. Для каждого из них указано, как он работает и какие пакеты с драйверами нужно установить. Информацию о сканерах можно найти на сайте [www.sane-project.org](http://www.sane-project.org). В дистрибутивах, в том числе и тех, о которых Вы говорите, есть хорошие утилиты для настройки устройств, но принтеры можно добавить и через браузер с помощью CUPS, стандартной системы управления драйверами принтеров с web-интерфейсом. Для добавления нового принтера или изменения существующих откройте в браузере адрес <http://localhost:631>.

## 5 Умиряющая сеть

**В** У моего Asus Eee PC – ему около четырех лет – проблемы с подключением к сети. Один мой добрый знакомый восстановил



➤ Web-браузер – все, что нужно для доступа к CUPS и документации.

вает сеть, но спустя день-два она пропадает снова. Я хотел бы переустановить Ubuntu 10.04 и посмотреть, изменится ли что-то, и если да – установить 12.04. Я бесплодно потратил несколько дней, пытаюсь скопировать файлы с LXF DVD159. Последняя моя проблема в том, что я нигде не могу найти и загрузить Dialog. Я копирую некоторые файлы на внешний жесткий диск и подключаю его к ноутбуку. Но какие файлы мне нужны на самом деле и как к ним обращаться?

Если это не помогает, стоит ли делать то же самое с 12.04? В ней должна быть обновленная версия Firefox. Или это аппаратная ошибка в Asus? Как это можно узнать?

Дэвид Аллен [David Allen]

Первый вопрос – что именно делает Ваш добрый знакомый? Если он таит это от Вас, то он не такой уж и добрый. Это похоже на аппаратную ошибку – драйвер беспроводной сети могло вывести из строя обновление ПО, но на таком старом компьютере это маловероятно – драйвер не должен был измениться так сильно. Кроме того, Eee PC – дешевый компьютер, и он недолговечен: у меня самого было два, и ни один четырех лет не прослужил.

Если не работает сетевое соединение, обновление браузера не поможет. Я бы попробовал несколько Live-дистрибутивов, желательно не на базе Ubuntu, и посмотрел, остается ли проблема в других дистрибутивах.

Если они запускаются без проблем, похоже, это программная ошибка, и Вам должна помочь установка более поздней версии (переустановка того же самого дистрибутива не поможет – это рабо-

тает только в Windows, и только в краткосрочной перспективе). Так мы переходим к следующей части письма: как записать образы CD или DVD на USB-брелок. Просто скопировать файлы нельзя: нужно перенести и загрузочные секторы. Простейший вариант для LXF DVD – воспользоваться скриптом `dvd2usb.sh`, как Вы уже узнали, но для этого нужна программа *Dialog*. Это стандартный пакет, который можно установить из менеджера пакетов любого дистрибутива.

Вы не написали, каким дистрибутивом пользуетесь, но если это Ubuntu, то *Dialog* находится в пакете *dialog*, который можно установить из *Software Centre*, *Synaptic* или из терминала командой:

```
sudo apt-get install dialog
```

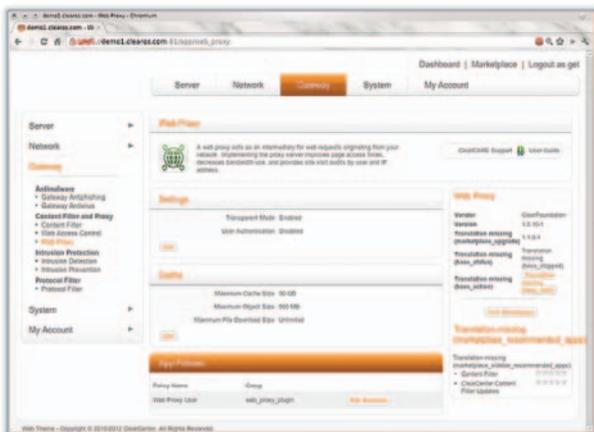
Существует и более общая программа для преобразования CD/DVD в USB – *Unetbootin* (<http://unetbootin.sourceforge.net>), которую также можно загрузить из *Software Centre*. С ее помощью можно преобразовать большинство Live-дистрибутивов в загрузочные USB-брелки.

## 6 Дистрибутив для прокси

**В** У меня есть старый 32-битный компьютер, и я хочу превратить его в прокси-сервер. Не могли бы вы посоветовать, каким дистрибутивом лучше воспользоваться?

Ян Флеминг [Ian Fleming]

На Ваш вопрос ответить непросто, так как Вы не указали назначения своего сервера. Если это web-прокси, то в любом дистрибутиве можно установить *Squid* (существуют и другие web-прокси, но *Squid* – самый популярный). Вам нужен просто web-прокси



» ClearOS превращает старый компьютер в удобный интернет-шлюз с web-прокси и многими другими сервисами.

с кэшированием или нужны и дополнительные возможности, наподобие фильтрации контента?

Существуют дистрибутивы, предназначенные именно для этой цели – превращения старого (или не очень) компьютера в интернет-шлюз с несколькими сервисами, в том числе с прокси-сервером или фильтром. Один из них – ClearOS, ранее известный как ClarkConnect. Его все еще можно найти на сайтах [www.clarkconnect.com](http://www.clarkconnect.com) и [www.clearfoundation.com/Software/overview.html](http://www.clearfoundation.com/Software/overview.html).

ClearOS – дистрибутив интернет-шлюза, в котором есть все, что Вам нужно, и многое другое. Возможно, «многое другое» Вам пока незачем, но приятно знать, что при необходимости мож-

но включить дополнительные опции – от антивирусного сканера до VPN-сервера и главного контроллера домена Windows. В основе ClearOS лежит CentOS, основанный на Red Hat Enterprise Linux, так что у него солидный фундамент.

Все управление осуществляется через web-интерфейс; после установки можно поставить сервер в уголок и отключить от него клавиатуру и монитор – для счастья ему будет достаточно питания и сетевого подключения. Чтобы понять, удовлетворяет ли он Вашим потребностям, воспользуйтесь любезно предоставленной разработчиками демо-версией. Откройте в браузере адрес <https://demo1.clearos.com:81/app/dashboard> и зайдите в систему с пользователем get и паролем clear. Выберите подходящий язык интерфейса – по умолчанию, видимо, включается последний использованный.

В разделе загрузок также содержатся образы виртуальных машин VMware и VirtualBox – это еще один хороший способ познакомиться с дистрибутивом перед установкой на реальный компьютер. Если Вы собираетесь попробовать систему на одном из бесплатных продуктов VMware, следует загрузить версию VMware Basic, а не VMware Enterprise. **LXF**

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, так как проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам нужно знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства. Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (<http://hardinfo.berlios.de/>) – она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файл, который вы сможете приложить к своему письму. Альтернативный и не менее удобный вариант – *lshw* (<http://ezix.org/project/wiki/HardwareLiSter>). Одна из этих программ должна быть включена в ваш дистрибутив (а то и обе).

Если вы не хотите или не можете их установить, выполните следующие команды в терминале от имени root и приложите файл **system.txt** к письму. Это здорово поможет диагностике.

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```



Часто задаваемые вопросы

# Удаленные рабочие столы

» Я хочу подключиться к домашнему компьютеру с работы и запускать на нем графическое приложение, например, почтовый клиент. Это возможно?

Да. Существует два основных способа решения вашей задачи и несколько вариантов для каждого из способов.

» И что это за способы?

Первый – перенаправление X, который лучше всего подходит для той ситуации, когда на обоих компьютерах установлен Linux. Он работает через SSH. В удаленной оболочке запускается программа, отображаемая на локальном рабочем столе. Например:

```
ssh -X me@my.home.computer kmail
```

Параметр **-X** означает, что ssh должна использовать перенаправление X.

» Это все, что нужно для запуска программы на локальном компьютере?

Строго говоря, программа запускается на удаленном компьютере, но ее окно отображается в локальной системе. Возможно, понадобится изменить параметр **X11Forwarding** на **yes** в файле **/etc/ssh/sshd\_config**.

» А если нужно отобразить весь рабочий стол? Можно ли это сделать, перенаправив X?

Да – открыв новый сеанс рабочего стола, который будет отображаться на локальном компьютере.

» Это не одно и то же? Если оставить программу запущенной и попробовать поработать с ней, то ничего не получится, верно?

Нет, для этого вам понадобится про-

грамма удаленного рабочего стола, как в том случае, если бы на локальном компьютере стояла Windows. Самый распространенный вариант – VNC (Virtual Network Computer – виртуальный сетевой компьютер). Это клиент-серверная система, в которой на удаленном компьютере запускается сервер VNC, а на локальном – клиент. На сайте [www.realvnc.com](http://www.realvnc.com) можно найти пакеты для всех распространенных ОС, и вы сможете подключиться к рабочему столу в Linux с компьютера с Windows или Mac и наоборот.

» Какая программа мне понадобится?

В KDE уже есть все необходимое, так как в ней есть RFB (Remote Frame Buffer – удаленный буфер кадра). RFB – протокол, используемый VNC и совместимыми системами. В дру-

гих дистрибутивах установите пакет VNC или TightVNC из репозитория.

» В чем разница между VNC и TightVNC?

VNC – версия-оригинал, TightVNC ([www.tightvnc.com](http://www.tightvnc.com)) – версия с оптимизацией для медленных соединений (с более сильным сжатием).

» Имеет ли значение ширина канала? Будет ли это работать через медленную связь?

Отображение рабочего стола и синхронизация изменений требует большой ширины канала, и модемное соединение не подходит. Подойдет широкополосное подключение, но помните, что какой бы высокой ни была входящая скорость ADSL-соединения, исходящая вряд ли превышает 384 кбит/с, поэтому лучше воспользоваться TightVNC. При необходимости уменьшите размер экрана и избавьтесь от всего способного увеличить объем передаваемых данных.



# 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.

MusE » expect-Lite » StressItOut » Poezio » Revelation » Functy  
» Super Grub2 » gShogi » Pax Britannica » PDF Cube » Calise

## Музыкальный секвенсор

# MusE

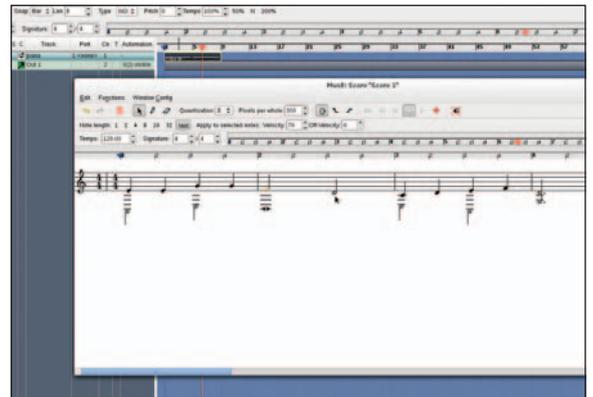
Версия 2.0 Сайт <http://muse-sequencer.org>

**Н**а рынке полным-полно музыкальных секвенсоров для Linux, однако о знакомстве с *MusE* определенно стоит задуматься. Если вы еще о нем не слышали, это, вероятно, из-за того, что он находился в своего рода спячке около двух лет, пока его авторы трудились над удивительным обновлением.

Для начала, в нем совершенно новый редактор партитуры и эмулятор DX11 – что явно взволнует некоторую часть аудитории – и полный переход на *Qt4*, способствовавший улучшению не только внешнего вида, но и качества работы. Есть

и менее заметные изменения; основное из них – то, что теперь *MusE* по умолчанию использует аудиосервер Jack. Таким образом, нужен сам Jack, и он должен быть запущен до того, как вы приступите к работе с *MusE*.

Стоит также учесть, что настройка Jack может оказаться заковыристой по час-



» Редактор партитуры – основное новшество, добавленное в *MusE 2*.

**«Теперь MusE по умолчанию использует аудиосервер Jack.»**

ти разрешений, и если вы не слышите от *MusE* звукового сигнала, вам в первую очередь, вероятно, стоит проверить именно их. При желании можно заставить программу вернуться к драйверам ALSA.

Для хранения треков *MusE* использует собственный формат файлов MED, и пока что вы не найдете ни примеров, ни демо-версий этого формата. Однако он также умеет импортировать стандартные файлы *.midi*, хотя они – что очевидно – будут отсутствовать в адаптациях под *MusE*; тем не менее, это неплохая отправная точка для ваших экспериментов.

*MusE* достаточно легко компилируется из исходника. В настоящее время он использует систему *Cmake*, так что вам придется проделать обычные

```
mkdir build
cd build
cmake ../
make
make install
```

Имеется несколько зависимостей – среди них и критически важные, и не очень. Если вы уже компилировали другие аудиопрограммы, они у вас, вероятно, есть, но в любом случае, сообщения об ошибках достаточно информативны.

*MusE* выбрали многие дистрибутивы, и к моменту, когда вы будете читать этот материал, вы уже сможете найти его в вашем обычном репозитории.

## Исследуем интерфейс MusE

### Меню

Меню очень обширны, но обратите исключительное внимание на меню Window.

### Элементы трека

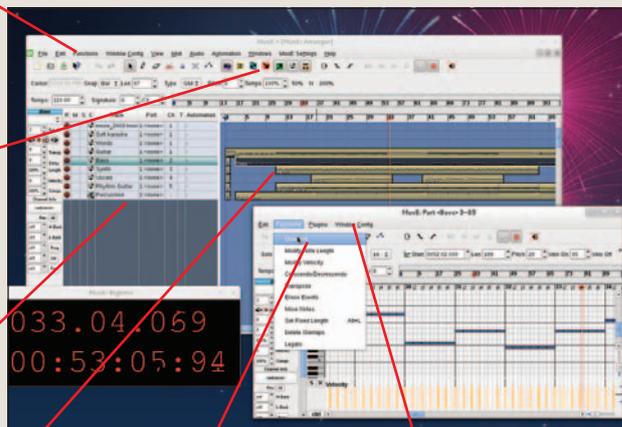
Эти обновленные значки помогут вам скрыть или показать определенные элементы текущего трека.

### Панель списка треков

Она показывает, какие инструменты вы загрузили.

### Расчет времени

Самая важная информация по синхронизации задается здесь.



### Редактирование

Можно редактировать отдельные части, дважды щелкнув по треку.

### Плагины

Здесь имеется множество плагинов и окон дополнительной информации.

## Системный инструмент

## expect-Lite

Версия 4.3.1 Сайт <http://expect-lite.sourceforge.net>

**Н**аписание системных скриптов должно быть не проблемой, а истинным наслаждением! Ну, хорошо, как минимум, оно не должно быть задачей, над которой приходится долго скрести в затылке. *Expect-Lite* получил свое название от достопочтенной командной утилиты *expect* от Дона Лайбса [Don Libes]. *Expect* была создана для преодоления часто встречающейся в системных скриптах проблемы – ожидание и обработки некоторой возвращенной величины от выполняемой команды. А при работе со всякими сетевыми заморочками это актуально как нигде и никогда. Как правило, *expect* сидит в канале **stdout**, в ожидании того, на что он и рассчитывает – ну там, пароля или подтверждения соединения.

*Expect-Lite* зашел чуть дальше, разработав среду скриптов. Вы, как и раньше, используете стандартные команды оболочки, но им предшествует индикатор `<` или `>`, чтобы указать, команда это ввода или

вывода. Время ожидания (указанное в виде цифровой величины в секундах после знака `@`) обеспечивает нормальную работу скрипта – он будет ждать ровно столько, сколько надо, таким образом избегая каких-либо проблем. Обычный скрипт может выглядеть примерно так:

```
>ssh root@host-021
<assword:
>>secret_password
# issue a command once logged in
>ls
>exit
```

Запуск скрипта будет по-прежнему перекидывать выходные данные неуловленных команд в оболочку, и вы сможете отслеживать все, что происходит. Факти-

«**Expect-Lite зашел дальше, разработав среду создания скриптов.**»

```
File Edit View Search Terminal Tab Help
Terminal
: == List screens and capture screen references
>screen -list | grep pts | sort | head -1
$myscreen1=[(0-9)+.pts.[0-9]+.[a-z]- 0-9]+)
>screen -list | grep pts | sort | tail -1
$myscreen2=[(0-9)+.pts.[0-9]+.[a-z]- 0-9]+)
: == Enter screen2, ping and sleep, then detach
>screen -r $myscreen2
# Inside existing screen, kick it to get a prompt
>^M
>ping -c 2 localhost
+packet_tx=[(0-9)+] packets transmitted
>sleep 2
>^M
>^D
<detached
: == Show packets transmitted inside screen: packet_tx=packet_tx
: == Kill all those screens and clean up
!killall screen
ps ax | grep screen
>
```

► Полагаем, вы сочтете этот инструмент для создания скриптов довольно удобным и полезным.

чески, есть даже специальный комментарий, который также выводит данные в **stdout**. Помимо подобных простых вещей, вы также сможете создавать внутри своего скрипта циклы и функции. Это достаточно просто, и более чем понятно описано в документации. *Expect-Lite* не прописан вам по жизненным показаниям. Он не делает ничего такого, чего нельзя сделать в *Bash* через родительскую команду *expect*. Но если вы освоите синтаксис, он сможет делать такие скрипты быстрее и проще для понимания.

## Системный инструмент

## StressItOut

Версия 0.1 Сайт <http://bit.ly/NAOfcT>

**Е**сть множество инструментов, пригодных для тестирования производительности оборудования, по какой бы причине вы ни решили ее проверить. Тесты памяти, тесты накопителей и задачи, нагружающие CPU, могут показать, как работает система и способна ли она обрабатывать определенные процессы, или даже просто работать так долго, как это необходимо для выполнения определенной задачи.

*StressItOut* несколько иной. Его основная идея в том, чтобы поместить систему в максимально стрессовую ситуацию, как и в большинстве тестов производительности. Но здесь идея в тестировании не основных компонентов системы, а защиты оборудования – т. е. очень хорошо, что ваш процессор выполняет столько операций в секунду, но сумеет ли он делать это до бесконечности, не надорвав систему охлаждения?

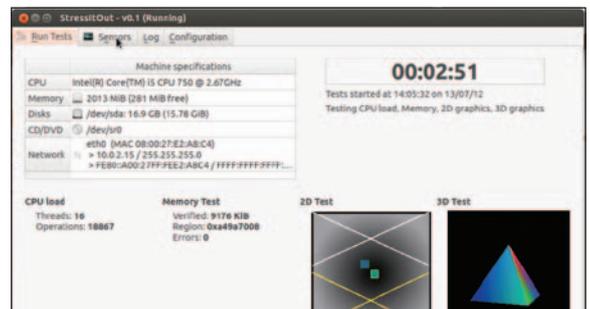
*StressItOut* подключается к датчикам, чтобы отслеживать температуру (и все

прочее, измеряемое датчиками) в системе, подвергая в то же время стрессовому воздействию все, что пожелаете. Это отличный способ проверить, работает ли этот новый кулер, который вы установили. Индивидуальные тесты можно включать и выключать через Preferences, а система отображения во вкладках позволяет легко переключаться между панелью управления тестированием, выходной информацией с датчиков и системным журналом.

Покамест этот инструмент простоват, но уже весьма полезен. И может стать еще полезнее при некоторой обратной связи и помощи в тестировании.

Вам кое-что понадобится для компиляции этой системы из исходника. Убедитесь, что у вас есть обновленная вер-

«**Покамест инструмент простоват, но уже весьма полезен.**»



► Единственный значимый результат тестирования – то, что ваш компьютер не спалится.

сия компилятора g++ (она у вас должна быть); а еще понадобятся инструменты разработки для Qt4 (обычно установка пакета **qt4-dev-tools** или подобного ему решит проблему). Программа также обращается к библиотеке *libsensors*, и если у вас возникнут ошибки компиляции, стоит проверить у себя наличие их установленной версии разработки. Процесс компиляции использует **qmake**:

```
qmake -makefile StressItOut.pro
make
```

После этого файл должен скомпилироваться. Здесь нет опции Install – просто запустите тест прямо из текущей директории.

Программа обмена сообщениями Jabber

# Poezio

Версия 0.7.5 Сайт <http://poezio.eu/en>

**I**RC – это здорово. Так здорово, что люди продолжают попытки изобрести его заново. Что, возможно, не очень умно. И это несмотря на наличие нескольких сервисов для обмена сообщениями, которые также превосходны – например, XMPP, или Jabber, поскольку он вам, вероятно, знаком лучше. Этот протокол обмена сообщениями поддерживает самые разные типы расширений, так что вы сможете легко участвовать в групповых обсуждениях, или, например, в обмене файлами. Ну прямо как, э-э, в IRC.

Что ж, авторы *Poezio* решили дать сдачи. Эти люди с их симпатичными новыми протоколами обмена сообщениями, вероятно, уже более популярны, чем IRC, хотя и не в состоянии похвастаться той же простотой.

*Poezio* – клиент Jabber, который работает в окне терминала. Он преобразует функции, общие для клиентов обмена сообщениями, в набор нажатий клавиш и команд, знакомых всем имевшим дело с IRC.

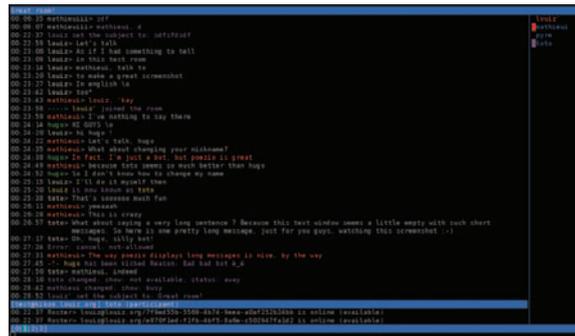
Так, например, список многопользовательских общественных чатов на сервере выведется просто по команде

```
/list server.name
```

Результат всего этого таков: как бы вы ни общались с людьми, через *Gajim*, или *Pidgin*, или даже *Google Chat*, все это будет совершенно так, как в IRC.

Естественно, отличия имеются, и эквивалент команды IRC есть не у всего, но в целом среда достаточно хорошо поддается настройке и имеет исключительно приличную документацию, так что вы легко сможете преобразовать все по собственному вкусу.

Это не просто примочка – *Poezio* очень быстр и удобен в использовании, а возможность переключаться между вклад-



Верните современный чат к его истокам с помощью *Poezio* и Jabber, сделав его чуть более эффективным.

ками, отображающими разные каналы общения, вероятно, несколько эффективнее попыток найти нужное окно на рабочем столе Ubuntu.

Однако, прежде чем запустить его, стоит потратить некоторое время на редактирование файла настройки. Иначе вы не увидите дальше создания анонимной учетной записи на тестовом сервере – уже неплохо, но было бы намного полезнее и удобнее иметь настоящую учетную запись Jabber там, где с ним можно связаться. Компиляция *Poezio* не принадлежит к самым простым, но есть подсказка – читайте файл README!

**«Poezio — клиент Jabber, работающий в окне терминала.»**

Менеджер паролей

# Revelation

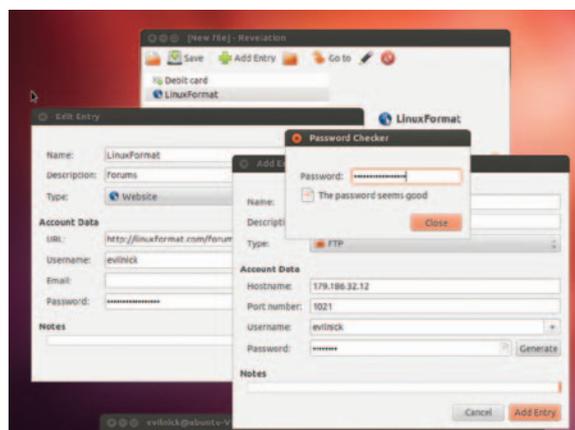
Версия 0.4.14 Сайт <http://revelation.olasagasti.info>

**Н**и дня не проходит без сообщений о том, что некий сайт лишился всей своей пользовательской базы данных. В некоторых случаях они настолько плохо зашифрованы, что воры за сутки получают все имена пользователей и пароли. А это проблема, особенно для тех, кто использует один и тот же пароль для всех сайтов – или его простые вариации. Что вполне объяснимо, поскольку – посмотрим правде в глаза – вы ведь вряд ли будете запоминать случайный ряд из 20 символов для каждого сервиса (кстати, этот трюк насчет *fellwifesubstanceexcite*, конечно, лучше, чем ничего, но у обычного английского текста энтропия очень низкая).

Итак, секрет в том, чтобы создать действительно случайный пароль, нечто вроде `gUJTt73gtyu;-;`, а потом найти то, что будет его для вас помнить. Если сайт взломают, один-то пароль вы потеряете, зато прочие ваши учетные записи будут в безо-

пасности. *Revelation* не только сохраняет ваши пароли, но и оценивает их на «взламываемость», и даже умеет генерировать для вас новые. Хранение и доступ к паролям (а также и PIN-кодам) осуществляется через интерфейс в виде списка. Каждый пароль может принадлежать к ряду различных категорий, что не только облегчает сортировку, но и дает возможность своего рода контекстного использования – можно, например, щелкнуть по веб-паролям, чтобы вас перевели на соответствующую веб-страницу в вашем браузере по умолчанию.

Здесь как раз и кроется некоторая проблема приложения – в нем нет прямой интеграции с веб-браузером (что, честно го-



*fellwifesubstanceexcite* – все же не самый лучший пароль.

воря, реализовать не так-то просто), и вам придется вырезать и копировать – не так удобно, как при управлении паролями самим браузером, но зато... случилось ли вам терять все данные браузера? Иметь резервную копию всегда на пользу.

Все данные, сохраняемые программой, подвергаются шифрованию высокого уровня, и пока вы помните главную ключевую фразу, они защищены даже лучше, чем ваше жилище. Главное, чтобы никто не взломал ваш компьютер на физическом уровне.

**«Каждый пароль может принадлежать к ряду различных категорий.»**

Программа для 3D-графики

# Funcy

Версия 1.25 Сайт <http://funcy.sourceforge.net>

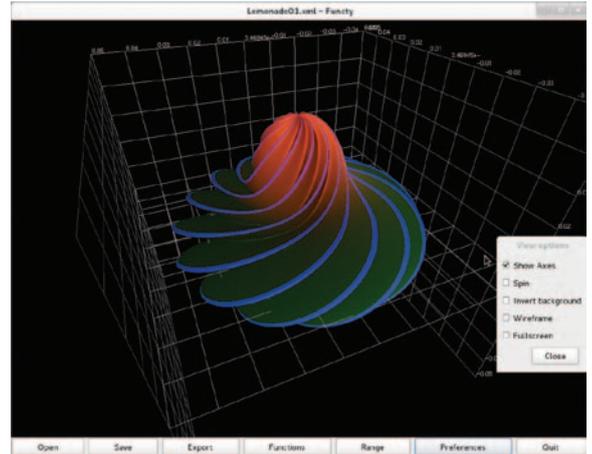
**Н**екотрые воображают, что стоит им глянуть на уравнение или таблицу с цифрами, и они представят себе трехмерную картину, описанную ими. Что ж, им повезло. Готов поспорить, им есть что порассказать на вечеринках. А для остальной части вселенной нужна какая-то программа визуализации, и если вам нужно что-то визуализировать, зачем мелочиться? Берите *Funcy*!

В отличие от львиной доли более, э-э, академических программ, *Funcy* не считает зазорным отображать вещи ужасающе и восхитительно реалистично. По этой причине здесь активно используется *OpenGL* для раскраски и рендеринга фигур, причем настолько детальных, что фигуры выглядят реальными физическими объектами.

Конечно, *Funcy* являет собой нечто большее, чем просто игрушку для генерирования этих завлекательных красок. С ее помощью можно сделать множество ценных вещей. Но – да, большую

часть времени вы потратите на возню с созданием всяческих эффектных зрелищ. Все это еще более впечатляет, когда вы обнаруживаете, что вашим уравнениям не обязательно быть статичными – они могут включать функцию *t*; встречаем анимацию!

Помимо исходного кода *Funcy*, вам понадобится скомпилировать библиотеку символов (с того же сайта); и еще нужен обновленный пакет *OpenGLtype*. В преамбуле настройки не учтена одна вещь – библиотека расширений *GLee*, которая не всегда устанавливается по умолчанию, так что проверьте ее наличие, если у вас появятся ошибки компиляции. Автор сообщает, что ему даже удалось успешно скомпилировать и запустить ее в Raspberry



» *Funcy* превращает генерирование приглядных объектов в развлечение.

Py Pi на Debian/ARM, так что у вас не должно быть никаких проблем, каким бы оборудованием вы ни пользовались.

Весьма необычным для подобного рода программ является отличная подборка файлов-примеров, что значительно облегчает освоение программы и, естественно, позволяет осознать, на что способна *Funcy*. Сохраненные файлы – это просто XML, и, честно говоря, иногда проще сгенерировать нужные объекты, непосредственно их написав, чем щелкать по бесконечным диалоговым окнам в самой программе. Но, возможно, это сугубо мое личное.

«Автору удалось скомпилировать и запустить ее в Raspberry Pi.»

Утилиты

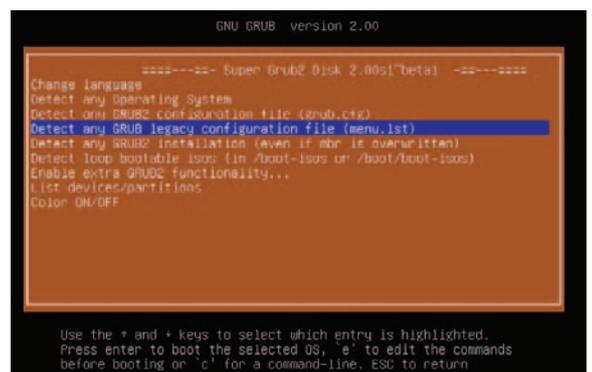
# Super Grub2

Версия 2 Сайт <http://bit.ly/dMDC1h>

**В**озможно, у вас никогда не было проблем с загрузкой Linux – по крайней мере, таких, что требовали более радикальных мер, чем выбор более ранней версии ядра в меню загрузки. Однако иногда система начинает сбивать, особенно если вы добавили новые диски или установили несколько версий Linux на одном компьютере (или создали разделы Windows или Mac) и хотите запустить одну из них. *Grub 2* – программа загрузки, которая решает такие проблемы за вас, а *Super Grub2* – загрузочный диск удобного размера, позволяющий вам поиграть с *Grub 2*. Мы должны остеречь вас от того заблуждения, что *Super Grub2* является новой версией превосходного диска *Super Grub disc*. Это одновременно и верно, и неверно. Программа загрузки *Grub* некоторое время назад перешла в версию 2, будучи значительно переписанной, что не только придало ей уйму новых функций, но и разорвало связь

с более ранними версиями *Grub*. Есть множество дистрибутивов, которые используют одну из этих версий – либо ни одной. Диск *Super Grub* по-прежнему существует, но *Super Grub2* создан на базе этой новой версии программы-загрузчика.

Если вы хотите применять его в качестве спасательного диска, знайте, что использование им новой версии *Grub* не означает, что нельзя загрузить более старые версии систем или другие системы, например, Windows, с его помощью. Прелесть *Grub* в том, что у него собственный набор команд для разных задач, например, для доступа к образам ОЗУ или выживания загрузочного кода с видимых им разделов. Это также отличная среда для тести-



» *Super Grub2* – инструмент, который заставит ваш компьютер нормально работать.

рования загрузочного кода при использовании нестандартного оборудования и для изучения работы *Grub 2*. Он безопасен – в отличие от старого диска *Super Grub*, новая версия не может даже ничего записать в MBR; она просто работает.

Эта новая версия загружается прямо с диска этого месяца, но можно и скопировать ISO на носитель USB.

«Super Grub2 создан на базе новой версии загрузчика.»



Игра сёги

# gShogi

Версия 0.4.3 Сайт <http://bit.ly/akZMFS>

**С**ёги [Shogi] – это традиционная японская игра, похожая на шахматы. Сходство в том, что у вас есть фигуры, стоящие по обе стороны доски, и цель заключается в захвате вражеского короля. Однако различий здесь, возможно, больше, чем сходства. Особенно в том, как ходят фигуры. Есть и несколько других странных правил.

Короче говоря, это очень запутанная игра, и вы потратите первый десяток игр, разгромно проигрывая на первых ходах любому оппоненту, имеющему представление о происходящем. Но выучите фигуры и ходы, и она станет для вас не просто свежей альтернативой шахматам, а тонкой и хитрой игрой.

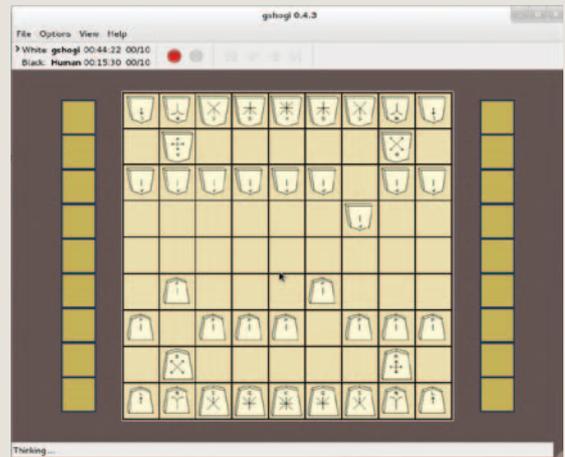
В Linux уже есть набор движков и интерфейсов для игры в сёги, и *gShogi* строится на нем. По большей части он основан на программе *GNUShogi*, в ко-

торую можно неплохо играть, но *gShogi* имеет ряд улучшений.

Изменяемый размер игрового поля делает ее более приличествующей для помещения на современных настольных ПК, и она может сохранять и загружать игру в различных форматах. Здесь также есть возможность выбора игровых движков сёги – к счастью, интерфейс у них стандартный, и *gShogi* управится со всем, что поддерживает Universal Shogi Interface (USI).

При движении на C и симпатичном интерфейсе, написанном на Python/GTK, *gShogi* объединил лучшее из обоих миров для настроившихся поиграть. Установка про-

**«В Linux уже есть набор движков и интерфейсов для сёги.»**



► Подсказка для новичков: переключитесь на «западные» фигуры, чтобы понять, как они ходят.

граммы проста, если использовать инструменты Python, но вначале необходимо выполнить компиляцию:

```
sudo python setup.py build
```

```
sudo python setup.py install
```

Набор движков для этого клиента и инструкции по установке вы найдете на [www.johncheetham.com/projects/gshogi/usi.shtml](http://www.johncheetham.com/projects/gshogi/usi.shtml).

Экшн

# Pax Britannica

Сайт <http://paxbritannica.henk.ca>

**И**сходя из названия, вы могли бы представить некую основанную на торговле стратегию сражений за скудные остатки ресурсов в отдаленных краях, с боевыми броненосцами и стим-панковским вооружением. И ошиблись бы. Да, это стратегия, но совершенно иного сорта.

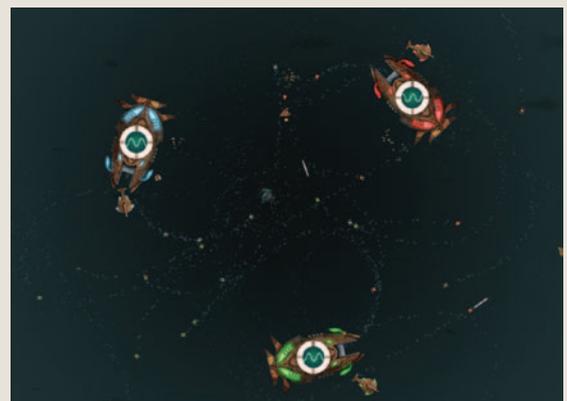
Идея заключается в том, что вы и другие игроки, числом не более 5, управляете своего рода подводным оружейным заводом. Завод может производить один из трех типов устрашающих (для врагов, а не для зрителей) кораблей, и у каждого из них свое особое оружие и возможности. Для их постройки нужно добыть ресурсы, что требует времени. Желая что-то построить, вы просто нажимаете одну-единственную клавишу, необходимую для игры, и отпускаете ее, когда визир попадает в нужный сектор.

Элементы стратегии зависят от того, захотите ли вы потратить больше времени, чтобы построить более мощные корабли (потенциально оставляя свой завод уязвимым для атак) или создать множество мелких кораблей, которые обеспечат некоторую защиту и смогут причинить некоторый урон противнику.

Для пушек сложности, вы можете добыть больше ресурсов и улучшить свой завод, чтобы в будущем производить ресурсы еще быстрее.

В режиме игры для двух игроков вы, скорее всего, мигмом ухватите суть игры и запросто победите искусственный ин-

**«Испытанием станет сражение с несколькими игроками.»**



► Это как «Хороший. Плохой. Злой», только под водой и со сложным автономным вооружением.

теллект. Реальным испытанием станет сражение с несколькими игроками, когда время поджигает сильнее, а результат непредсказуем, ибо зависит в основном от того, где ваше автономное судно решит напасть. Режим же игры для четырех игроков вполне способен взорвать мозг.

Хорошая графика, прекрасный саундтрек и отличная играбельность в многопользовательском режиме заставят вас обращаться к этой игре снова и снова. Поверьте нам.

## Утилиты

## PDF Cube

Версия 0.0.5 Сайт <http://code.100allora.it/pdfcube>

Что может сделать ваш и без того превосходно сделанный документ PDF еще круче? Его отображение на 3D-куб! Это настолько очевидно – даже странно, что никто до этого не додумался раньше.

О рендеринге PDF заботится библиотека *Poppler*, используемая всеми остальными программами, которые управляют файлами PDF, так что не переживайте насчет экономии на внешнем виде и точности документа. Кубический эффект требует наличия OpenGL. Она хорошо работает на оборудовании NVidia на моей машине, но ваши результаты могут быть иными. Понятно, что она будет работать лучше с более быстрой видеокартой, но даже старые 3D-карты должны суметь нормально отобразить вращающийся куб.

Эта программа разработана не просто как обычная программа просмотра PDF, а, скорее, как изящный визуальный фокус. Она нацелена на работу с альбомными PDF размера A4. Вы можете предоставить ей другие PDF, и она их обработает,

но результат может отличаться от ожидаемого вами. Как видно по номеру версии, это ранний релиз кода, и автор считает, что он находится на альфа-стадии, так что пока не стоит устраивать на нем важные корпоративные презентации. Но все же это неплохая программка, и она оживит скучную презентацию.

Компиляция довольно проста (понадобятся библиотеки boost), а если вы работаете в Ubuntu, вы обнаружите, что там уже есть пакет для **pdfcube**, который, вероятно, будет обновлен до этой самой последней версии в ближайшем времени.



» Это как PDF, но в кубе!

## Утилита монитора

## Calise

Версия 0.4 Сайт <http://calise.sourceforge.net/wordpress/>

Истинные профи кодируют вслепую. Хотя это бывает небезопасно, особенно если на вашем столе выстроены горячие напитки – я счет потерял случаям, когда окунал CD в кофе. Но это все же лучше, чем когда я норовил отформатировать диетическое печенье.

Проблема офисов в том, что фоновое освещение изменяется, и во избежание перенапряжения зрения вам, видимо, надо будет подстраивать яркость монитора так, чтобы она соответствовала освещению.

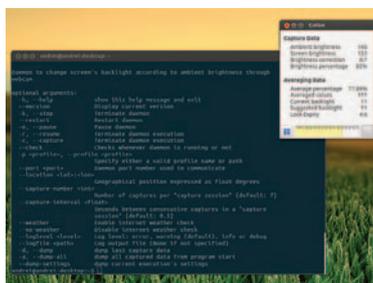
*Calise* работает, управляя вашей камерой. Периодически она будет ее применять для измерения фонового освещения и соответствующей настройки яркости вашего монитора. В идеале это делается очень постепенно, по мере изменения освещения.

*Calise* сидит в DBUS, и по вашим широте и долготе хитроумно выясняет время восхода и заката (просто используя библиотеку *pyephem*, но выглядит это как чу-

до). Зачем? А затем, что именно тогда освещение меняется быстрее всего.

Возможно, подстройка яркости монитора под внешнее освещение и не стоит у вас в списке неотложных дел, но попробуйте – и вас удивит перемена к лучшему.

*Calise* – довольно сложная смесь скриптов Python и кода C, с рядом зависимостей, которые вам, видимо, придется искать для своего дистрибутива. К счастью, львиную долю этого поясняет документация. **LXF**



» Ежеминутные изменения яркости рабочего стола отобразить довольно сложно, поэтому вот вам изображенные утилиты системного лотка.

## Также вышли

Новые и обновленные программы, тоже достойные внимания...

» **Equake 1.0**

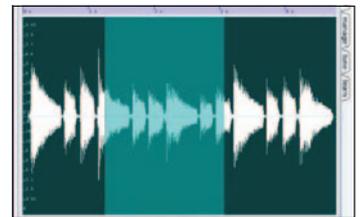
Своевременные предупреждения из системного лотка о сотрясениях. [www.e-quake.org](http://www.e-quake.org)

» **perltidy 20120701**

Украстье свой код Perl самым простым способом. <http://perltidy.sourceforge.net>

» **eko 0.7.1**

Зачем все усложнять? *Eko* – простой и эффективный звуковой редактор. <http://eko.sourceforge.net>



» Простая программа для создания звуковых сэмплов ищет... пользователей.

» **gfeedline 1.6.1**

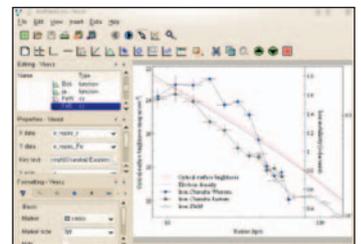
Обновление к клиенту Twitter, о котором мы недавно рассказывали. <http://code.google.com/p/gfeedline>

» **Gnome Partition Editor 0.13**

Не изменяйте размер разделов вашего жесткого диска, не обзаведясь самой последней версией! <http://gparted.org>

» **Veusz 1.16**

Научный пакет для вывода графиков в виде готовых к печати файлов PDF. <https://launchpad.net/veusz>



» Пусть все точки улягутся на симпатичный график с помощью *Veusz 1.16*.

» **psensor 0.7.0.1**

Графический монитор для критически важных датчиков материнских плат. <http://wpitchoune.net/blog/psensor>

» **LRZip 0.614**

Серьезное исправление самой хрупкой утилиты архивирования. <http://ck.kolivas.org/apps/lrzip>

# На диске

Пробуйте новую операционную систему уже сегодня!



**И**ногда бывает неплохо на шаг отступить и оценить картину в целом. Здесь, в Башнях LXF, мы стараемся донести до вас самые новые технологии Linux, а значит, предпочитаем выбирать для диска самые свежие и лучшие программы, а не самые популярные. Но в этом номере мы решили восстановить равновесие и включили всю десятку лучших по версии Distrowatch (плюс еще

несколько), и это, как нам кажется, создает взвешенную панораму программ Linux, которые применяются большинством.

В соответствии с данными Distrowatch, «горячая десятка» последние полгода выглядит так (в порядке убывания популярности): Mint, Ubuntu, Mageia, Fedora, Debian, OpenSUSE, Arch, CentOS, Puppy и PC LinuxOS.

## Лучшие дистрибутивы

# Горячая десятка

**П**ри всей рекламе, лишь один из десяти использует Gnome 3 по умолчанию (Fedora), а два напрямую держатся за Gnome 2 (CentOS и Debian). Подавляющее большинство (7) использует традиционный рабочий стол; то есть болтовня о вновь изобретенной парадигме рабочего стола – некоторое преувеличение. Возможно, всего удивительнее – их непохожесть: ведь все они черпают большую часть кода из одного источника СПО. Самая популярная среда рабочего стола (KDE) на первом месте только в трех.

Конечно, это только рабочие столы по умолчанию. В большинстве дистрибутивов их можно изменить одним нажатием кнопки. Где же истинное различие? В них – разные менеджеры пакетов, хотя на самом деле их функции весьма сходны.

Спустя более десятилетний опыт использования Linux и шести месяцев подбо-

Рис. 1. Чтобы выбрать идеальный для вас дистрибутив, просто соедините свое оборудование со своими запросами по стабильности.



правилу, предназначены для новых машин; но это не повод старым машинам работать на старых программах.

Эти два различия можно применить для создания того, что мне нравится называть Матрицей Эверарда для сравнения дистрибутивов Linux (рис. 1). Она не принимает в расчет приложения по умолчанию, которые легко изменить после установки (за исключением тех случаев, когда они привязаны к одной из двух осей – например, *Gnome Office* вместо *Libre Office*).

Так, CentOS ставит стабильность выше инноваций, и работает на достаточно медленных машинах. А вот Puppy использует более современное ПО, но во главу угла ставит эффективность. Все дистрибутивы на базе KDE, по моему ощущению, нацелены на более быстрое оборудование, и поэтому разместились с правой стороны графика. Конечно, размещение каждого дистрибутива по осям не вполне научно, а выбор осей исключительно субъективен.

## «Дистрибутивы на базе KDE, по-моему, для быстрого оборудования.»

ра дистрибутивов для LXF DVD я пришел к выводу, что подлинных различий между дистрибутивами только два: насколько они передовые и на какое оборудование нацелены. В свежих версиях ПО почти всегда больше ошибок, чем в проверенном и протестированном коде, так что новейшие функции достигаются за счет стабильности. В одних ситуациях это окупается, в других – нет.

Второе различие отчасти связано с первым: новые программные продукты, как

## Важно ВНИМАНИЕ!

Прежде чем вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензию.

### Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу [disks@linuxformat.ru](mailto:disks@linuxformat.ru), для получения содействия.

Компьютеры должны быть удовольствием

# Doudou

**М**аленькие дети и компьютеры ладят не всегда. Настольные системы, созданные для обеспечения производительности, не слишком подходят для незрелых умов. Цель Doudou – исправить этот недостаток. Это среда live, запускаемая на любом компьютере. Режим live означает, что жесткие диски не подмонтируются, и можно позволить малышам щелкать, где угодно, не бо-

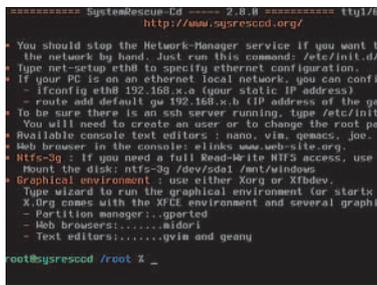
ясь, что они случайно удалят важные файлы. Doudou содержит только программы для детей, а не джентльменский набор для системы Linux общего назначения. Для работы в Интернет предусмотрен *Epiphany*, но со включенной по умолчанию фильтрацией контента. *Childsplay* и *GCompris* – пакеты образовательных игр. Помните, что это – система live, и все, что вы решите сохранить, следует помещать на USB-брелок.



► Подборка игр и безопасный web-браузер – все, что нужно ребенку в дистрибутиве Linux.

Когда хорошему компьютеру поплохело

# System Rescue CD



► В редакции LXF System Rescue CD и *ntfsundelete* недавно вернули из небытия случайно удаленную музыку.

**Г**оворят, что три вещи в жизни неизбежны: смерть, налоги и потеря данных. Перекопав весь Интернет, мы не смогли найти свободного ПО, способного справиться с первыми двумя, но этот дистрибутив live, возможно, сумеет помочь вам справиться с третьей. Он предназначен только для тех случаев, когда с вашим компьютером случается некая проблема (и поможет восстановить как Windows, так и Linux). По умолчанию System Rescue CD не загружа-

ется в графической среде. Если вам удобнее работать с мышью, введите **wizard**, чтобы запустить *Xfce*. Этот дистрибутив не чинит ваш компьютер волшебным образом, но дает инструменты для создания и восстановления резервных копий, восстановления случайно удаленных данных (в зависимости от файловой системы) и считывания информации с поврежденных жестких дисков. В общем, это отличный дистрибутив для лежания на полке – так, на всякий случай.

Контролируйте свой жесткий диск

# Parted Magic

**Е**сли вы используете только одну ОС, вам, возможно, никогда не придется ломать голову по поводу разбиения жесткого диска на разделы: просто выберите установки по умолчанию при установке, да так и оставьте. Однако если вам хочется иметь несколько установленных ОС или нестандартную разметку диска, вам, вероятно, надо в большей степени взять на себя управление тем, что происходит. Большинство ди-

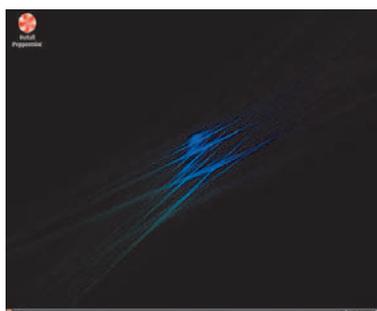
стрибутивов предлагают поддержку деления диска на разделы во время установки, но в ней может не хватать функциональности. Parted Magic также содержит другие инструменты, которые могут пригодиться при работе с жестким диском: например, инструмент для безопасного удаления данных с диска и инструмент для клонирования дисков. Короче говоря, это универсам, где вы найдете все необходимое для работы с жестким диском.



► [partedmagic.com](http://partedmagic.com) содержит серию документов и видео в помощь управлению жестким диском.

Множество облачного добра

# Peppermint



► Причина оголения этого рабочего стола в том, что все ушло в облако.

**О**блачные вычисления разведали нас по лагерям почти так же, как и горячие споры в мире Linux, о которых говорилось ранее в этом номере. Если вы – заклятый враг облачных технологий, вы, возможно, решите зажмуриться и притвориться, что данного материала нет. Но для людей с открытым складом ума этот дистрибутив стоит внимания. Он включает легковесный менеджер окон, web-браузер и несколько инструментов для взаимодей-

ствия с облаком. Например, офисным пакетом здесь является Google Docs. Однако отсюда следует, что дистрибутив подходит только для компьютеров с доступом к Интернету. Основная система достаточно нетребовательна к ресурсам и должна хорошо работать и на старых компьютерах, но это, конечно, зависит от web-приложений, которые вы планируете использовать: они с удовольствием глотают и время CPU, и память. LXF

## 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](http://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><b>LXF159</b><br/>Июль 2012</p> <ul style="list-style-type: none"><li>» <b>Linux против Windows</b> Если слон на кита налезет, кто кого соберет?</li><li>» <b>Рулим обновлениями</b> Менеджеры пакетов – без них никуда</li><li>» <b>OpenCL</b> Открытый стандарт для параллельного программирования</li><li>» <b>Презентации</b> Как обаять аудиторию и заразить своей идеей</li></ul> <p><b>LXFDVD:</b> Ubuntu 12.04 LXF Remix и еще 3 дистрибутива, FWBuilder, LibreOffice 3.5.2 и прочее...</p> <p><b>Печатная версия:</b> <a href="http://shop.linuxformat.ru/lxf_159/">shop.linuxformat.ru/lxf_159/</a><br/><b>PDF-версия:</b> <a href="http://shop.linuxformat.ru/elxf_159/">shop.linuxformat.ru/elxf_159/</a></p> |  <p><b>LXF160</b><br/>Август 2012</p> <ul style="list-style-type: none"><li>» <b>Red Hat</b> Как заработать миллиард долларов</li><li>» <b>Общаемся в открытую</b> Социальные сети, уважающие приватность</li><li>» <b>OpenCL</b> Открытый стандарт для параллельного программирования</li><li>» <b>Кластер всем миром</b> Подключаем свой ПК к прогрессу науки</li></ul> <p><b>LXFDVD:</b> Mageia 2 и еще 3 дистрибутива, BCCD, Fotoxx, YAGF, руководство администратора Debian и прочее...</p> <p><b>Печатная версия:</b> <a href="http://shop.linuxformat.ru/lxf_160/">shop.linuxformat.ru/lxf_160/</a><br/><b>PDF-версия:</b> <a href="http://shop.linuxformat.ru/elxf_160/">shop.linuxformat.ru/elxf_160/</a></p> |  <p><b>LXF161</b><br/>Сентябрь 2012</p> <ul style="list-style-type: none"><li>» <b>Обновляйтесь!</b> Разумный выбор нового оборудования</li><li>» <b>Строим сайт</b> Системы управления контентом</li><li>» <b>OpenStack</b> Linux облачных вычислений</li><li>» <b>Moodle</b> Жаждете поделиться знаниями? Открывайте курсы!</li></ul> <p><b>LXFDVD:</b> ROSA Marathon 2012, Linux Mint 13 и еще 2 дистрибутива, Phoronix Test Suite, DigiKam, Packet Fence, видеоурок по работе с микрокомпьютером МК802 и прочее...</p> <p><b>Печатная версия:</b> <a href="http://shop.linuxformat.ru/lxf_161/">shop.linuxformat.ru/lxf_161/</a><br/><b>PDF-версия:</b> <a href="http://shop.linuxformat.ru/elxf_161/">shop.linuxformat.ru/elxf_161/</a></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](http://www.linuxcenter.ru) или [shop.linuxformat.ru](http://shop.linuxformat.ru) получают электронную версию в подарок!

Подписывайтесь на журнал на [www.linuxformat.ru/subscribe/](http://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 2011 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](http://shop.linuxformat.ru)



6 месяцев  
**1230** руб.

12 месяцев  
**2280** 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 этажом по списку; “tgz” – это сокращение от “tar.gz”;
- » **имя\_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя\_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя\_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя\_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя\_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

### Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: [disks@linuxformat.ru](mailto:disks@linuxformat.ru)

**Внимательно прочтите это перед тем, как использовать LXF DVD!**

**А ТАКЖЕ:** 9 книг о Linux, программы из раздела NoForks и многое другое

**А ТАКЖЕ:** Arch, Fedora, PClinuxOS и Puppy

**7**  
Хороших  
дистрибутивов  
2012 года

**Тоже достойны рассмотрения:**  
Debian, Doudou, LegacyOS, Rated Magic, Repermint, System Rescue CD, Ubuntu

**8**  
лучших  
дистрибутивов  
2012 года

**ЧЕМПИОНЫ ПО ВЕРСИИ LINUX FORMAT:**  
» Mint » Mageia » openSUSE » CentOS

Октябрь 2012  
**LXF DVD 162**

**LINUX**  
FORMAT

Октябрь 2012  
**LXF DVD 162**

**LINUX**  
FORMAT

Сторона 1

- ДИСТРИБУТИВЫ с загрузкой с LXFVD**  
**Arch** Легкий дистрибутив с гибкими возможностями настройки, 32- и 64-разрядные сборки
- CentOS 6.3** Enterprise Operating System, созданная сообществом на базе Red Hat Enterprise, 32-разрядная сборка
- Fedora 17 Live** Обновленная версия популярного дистрибутива, 32-разрядная сборка
- Magia 2 Live** Вторая версия ответвления Mandriva, 32-разрядная сборка
- Minix 13 Live** Сборка с рабочим столом Synaptik, 32-разрядная сборка
- OpenSUSE 12.1 Tumbleweed** Обновляемая из наиболее свежих репозиториях версия популярного дистрибутива, 32-разрядная сборка
- PC Linux OS 2012** Простой в использовании Live-дистрибутив, 32-разрядная сборка
- Puppy Slacko Live** Обновленная версия популярного Live-дистрибутива, 32-разрядная сборка

Сторона 2

- ДИСТРИБУТИВЫ**  
**Debian 6.0.4 Live** Актуальная версия популярного дистрибутива, 32-разрядная сборка (загрузка с LXFVD)
- Doudou 1.2** Дистрибутив для самых юных пользователей, 32-разрядная сборка (загрузка с LXFVD)
- LegacyOS 4** Легкий дистрибутив для устаревшего «железа» (ISO-образ)
- Parted Magic 2012-06-26** Live CD для работы с разделами диска (загрузка с LXFVD)
- Peppermint 3** Дистрибутив для работы с облачными службами Google и другими, 32-разрядная сборка (загрузка с LXFVD)
- System Rescue CD 2.8.0** Live CD для восстановления системы и данных, 32- и 64-разрядные сборки (загрузка с LXFVD)
- Ubuntu 12.04 Live** Актуальная версия популярного дистрибутива, 32-разрядная сборка (загрузка с LXFVD)

- ДОКУМЕНТАЦИЯ**  
**9 книг о Linux** (на английском языке):  
Bash Scripting  
 Bourne Shell Scripting  
 Cathedral Bazaar  
 Debian Administrator's Handbook  
 Intro to Linux  
 Linux Dictionary  
 Linux Kernel in a Nutshell  
 System Administrators Guide  
 GNU Tools Summary

- gShogi 0.4.3** Игра Shogi (японские шахматы), написанная на Python/PYGTK
- Muse 2.0** Графическая программа для записи и редактирования музыки
- RaxXnamic** Игра, пошаговая стратегия войны на море
- PDFcube 0.0.5** Программа просмотра PDF с большим количеством различных эффектов
- Poezio 0.7.5** Консольный клиент XMPP (Jabber)
- Revelation 0.414** Менеджер паролей для Gnome
- StressItOut 01** Программа для тестирования производительности компьютера (исходные коды)
- SuperGrub 2** Загружаемый диск для восстановления GRUB (ISO-образ и загрузка с LXFVD)

НОТРИСКС

- Calaise 0.3.0** (Camera Light Sensor) Утилита для определения освещенности и регулировки яркости монитора
- expedit-Lite 4.3.1** Средство автоматизации работы в командной строке
- Functy 0.25.0** Программа для построения 3D-графиков

**Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!**

**КОММЕНТАРИЙ** Приставьте ваши пожелания и предложения по электронной почте: [info@linuxformat.ru](mailto:info@linuxformat.ru)  
**ДРУГИЕ ДИСКИ** В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу [disk@linuxformat.ru](mailto:disk@linuxformat.ru)

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаритурный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принести на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с нашего сайта: [www.linuxformat.ru](http://www.linuxformat.ru)  
Тираж издательства 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 через *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке – например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor=0,3,0 12 16M
```

Первое слово в этой строке – метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root и выберите вкладку `Burn` и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на `Combust!` Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.

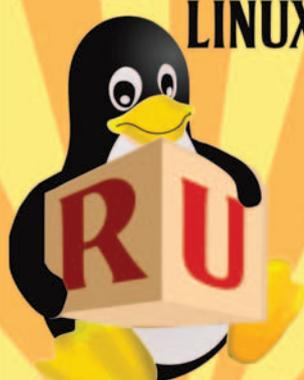
Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-нибудь с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.

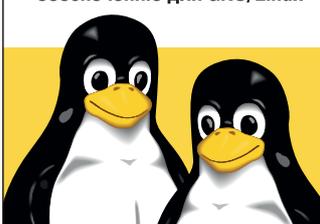
# ФОРУМ №1

ДЛЯ ВСЕХ ПОЛЬЗОВАТЕЛЕЙ LINUX



**LINUXFORUM.RU**

**LinSoft.info**  
Путеводитель по программному обеспечению для GNU/Linux



**WWW.LINSOFT.INFO**

**Linux по-русски**

Библиотека книг, статей и переводов о Linux

**WWW.RUS-LINUX.NET**

Аппаратно-программный комплекс

# DR.WEB OFFICE SHIELD

Комплексное решение задачи антивирусной и антиспам-защиты для малых и средних предприятий



88 679 руб.

**DR.WEB OFFICE SHIELD TWISTER**  
Рассчитан на 250 ПК



65 240 руб.

**DR.WEB OFFICE SHIELD NEO**  
Рассчитан на 50 ПК

**Dr.WEB®**

**Linux center**  
www.linuxcenter.ru

[WWW.LINUXCENTER.RU/SHOP/ANTIVIR/DR\\_WEB/](http://WWW.LINUXCENTER.RU/SHOP/ANTIVIR/DR_WEB/)

# LINUX FORMAT

Главное в мире Linux

## Как разместить рекламу в разделе Classifieds?

|   |              |
|---|--------------|
| 1/4 полоса (210 × 297 мм)               | 165 200 руб. |
| 1/2 полосы горизонтально (197 × 144 мм) | 88 500 руб.  |
| 1/2 полосы вертикально (102 × 278)      | 88 500 руб.  |
| 1/4 полосы вертикально (98 × 138 мм)    | 53 100 руб.  |
| Фотоблок (44 × 113 мм)                  | 15 000 руб.  |

**Тел.: +7 812 309 06 86**

Цены указаны с учетом НДС

**Linux center**  
www.linuxcenter.ru

*Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!*

**Широкая сеть представительств в разных городах позволит вам оптимизировать процессы логистики и доставки товара.**

Подробнее о партнерской программе:  
[www.linuxcenter.ru/partner/](http://www.linuxcenter.ru/partner/)

Международная **КОНФЕРЕНЦИЯ ВЫСТАВКА**

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

# ИТО

Место проведения:  
Москва  
Ленинские горы  
Московский государственный университет имени М. В. Ломоносова  
2-й учебный корпус  
дом 1, строение 52

**7-9 ноября**

<http://ito.su> email: [info@ito.su](mailto:info@ito.su)

Конференция проходит при поддержке:



**Unix Education Center**  
Россия, 190000, Санкт-Петербург  
Черноморский переулок, дом 4  
Тел.: + 7 (812) 611-1575



# Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



[WWW.LINUXCENTER.RU/SHOP/GIFTS/](http://WWW.LINUXCENTER.RU/SHOP/GIFTS/)

# Школа LXF

Спонсор рубрики  
**Mandriva.ru**  
разработчик  
дистрибутива  
EduMandriva  
[www.mandriva.ru](http://www.mandriva.ru)

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

## Сказ о борьбе кактусов с сибирской зимой

**Михаил Выграненко** сочувствует испытаниям замерзающих экзотов словом и делом и даже предлагает оранжерею для них.

Что остается от сказки потом –  
После того, как ее рассказали?  
В. С. Высоцкий. «Песня Кэрролла»  
из радиоспектакля «Алиса в стране чудес» (1974)



Наш  
эксперт

**Михаил**

**Выграненко** – идеолог свободы в мыслях и порядка в делах на ниве просвещения, каким бы парадоксальным ни казалось это сочетание. В миру – учитель-методист.

**В**от и наступила сибирская зима. Нет, я не о времени года и не о погоде, за окном-то все же сравнительно тепло. А помните песенку «Синоптики» группы «Наутилус Помпилиус»? В ней ведь тоже не про работников гидрометеослужбы, «сумевших выжить на лютом морозе». Я хочу вернуться к теме свободного программного обеспечения (СПО) в среде нашего школьного образования. Так ведь внедрение закончилось! – удивитесь вы. Действительно, об этом и сказ. И что же?

Государственный проект НП-12 отработан, выделенные деньги освоены, и все успешно отчитались. Славной фирмой «Альт-Линукс» создан и роздан в школы программный пакет<sup>1</sup>, открыты сайты поддержки<sup>2</sup>, катком по головам назначенных добровольцев прокатилось всеобщее форсированное обучение. Все. Пользуйтесь, учителя!

Но нет ли у вас ощущения некоторого уныния в этой завершенности? Так ведь это проект, любимое детище нашей административной системы! Эта схема отработана до мастерства, в советские времена она называлась кампанейщиной, и тоже практиковалась повсюду, хотя на словах зачастую порицалась. Теперь не порицается: теперь утверждается, что так и надо.

Время идет, минуло почти два года со дня завершения проекта. «Скоро сказка сказывается, да НЕ...» Сайты поддержки замерзли и НЕ обновляются, обученные НЕ несут света полученных знаний в массы, частично разбежались из школ, а большинство скрываются и, скорее всего, позабыли то, чему их учили. Так всегда бывает с назначенными добровольцами, предоставленными самим себе. Педагогические форумы и сообщества НЕ бурлят СПО-креативом, то бишь новой методикой и дидактикой. Руководству же всех уровней и его многочисленному окружению внедрение СПО изначально было обузой, о которой НЕ хочется вспоминать и уж тем более – включать в список текущих приоритетов и инноваций. И только неугомонные наши разработчики выдают на гора очередные дистрибутивы<sup>3</sup>, совершенствуясь без оглядки на учительскую обратную связь... Ну пользуйтесь же, учителя!

Это и есть зима. Суровая, сибирская. Зима в делах, зима в головах. Системного развития, планомерной заинтересованной

деятельности нет. Какой смысл ожидать положительного результата для школы? Есть только локальные корпоративные заинтересованности: кто-то получил деньги, кто-то отточил имидж, а кто-то издал публичный треск. Как, например, отечественное министерство просвещения... но ведь это уже давно стало его программой – выдавать желаемое за действительное. Учителя и особенно директора школ получили головную боль.

Хотя все-таки два результата есть. Всего два, но существенных. В школе педагогические работники теперь знают о правомерности использования ПО, и у них появился выбор. Раньше-то, скажут, никто понятия не имел о стоимости компьютерных программ, и что у них бывает автор (ну, или не хотел иметь понятия). И что есть правовые нормативы, устанавливающие все это. Все было в помыслах чисты, потому как не ведали. Включая тех, кому по долгу службы положено следить за порядком. Теперь не так. То есть за порядком по-прежнему никто не следит, но педагоги уже кое-что знают, и им, возможно, в глубине души неловко, совместно нарушать. Как-никак, они – педагоги. Но чтобы не нарушать, надо потрудиться, во-первых, в работе над собой, а во-вторых, в плавании против всеобщего течения на авось. Ведь нельзя же представить, что школы смогут оплачивать необходимое проприетарное ПО.

И вот сложилась занятнейшая ситуация умалчивания и выживания. Как дальше ветер подует? Будут наказывать или нет? Ведь так не хочется протянуть руку к диску со школьным пакетом СПО или скачать соответствующий дистрибутив в Интернете. В махровом феодализме школьного образования откуда взяться свободе, чтобы сделать осознанный и самостоятельный выбор? Зима крошечная. Холодно.

Холодно и вне стен школы. Нет, что-то, конечно, происходит. Вот международная конференция «Теория и практика использования свободного программного обеспечения» проведена в Барнауле<sup>4</sup>, Московский институт открытого образования продолжает дистанционно обучать по двум СПО-курсам учителей информатики<sup>5</sup>. Но отдельные мероприятия, проводимые силами энтузиастов, не делают погоду по стране. Российская ассоциация свобод-

ного программного обеспечения (РАСПО), используя площадку июньской конференции «Стратегия и тактика перехода органов государственной власти на использование свободного ПО (СПО)»<sup>6</sup> и обращаясь с письмом непосредственно к министру<sup>7</sup>, настойчиво твердит, что дело не в отечественных разработчиках, и не в обучении масс, и не в драйверах, иными словами – «дело не в деньгах и не в количестве женщин», как спел когда-то Б. Гребенщиков<sup>8</sup>, а в политической государственной воле прежде всего.

Лицемерная позиция и недалекость руководящих структур развращает население, оттачивает мастерство наведения тени на плетень повсеместно, замораживает все живые ростки энтузиазма и просто здравого смысла. «Реализация распоряжения № 2299-р фактически саботируется ответственными за его выполнение чиновниками» – констатирует РАСПО. Речь идет ни много ни мало о Распоряжении Правительства РФ от 17 декабря 2010 г. «О плане перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения (2011–2015 годы)»<sup>9</sup>.

А вот еще недавнее Постановление Правительства РФ № 343 от 18 апреля 2012 г. «Об утверждении Правил размещения в сети Интернет и обновления информации об образовательном учреждении»<sup>10</sup>, в котором устанавливается, что «технологические и программные средства, которые используются для функционирования официальных сайтов образовательных учреждений в сети Интернет, должны обеспечивать: а) доступ пользователей для ознакомления с размещенной на сайтах информацией на основе свободного и общедоступного программного обеспечения...». Это требование влечет за собой серьезную и объемную редакторскую работу для приведения цифровых ресурсов в соответствие, причем начиная с верхних штабов. В сочетании с тем, что в качестве стандарта документов для госструктур с 1 июня 2011 г. принят формат ODF (ГОСТ Р ИСО/МЭК 26300-2010)<sup>11</sup>, а в рамках направления «Национальная программная платформа» с 1 января 2012 г. вступил в действие стандарт «Информационные технологии. Свободное программное обеспечение. Общие положения» (ГОСТ Р 54593-2011) совместно с Правилами использования (Р 50.1.077-2011)<sup>12</sup>, казалось бы, работа на местах должна кипеть. Ан нет, тишина. В районных и муниципальных отделах образования, а тем более в школах ничего об этом не знают. Указаний нет. И кипения нет. Холодно.

Зима лютует. Или, о том же можно сказать словами *генерального директора компании «ПингВин Софт» Д. Комиссарова*: «Все мы хорошо знаем, что примерно 70 % ПО в нашей стране бесплатно. Точнее – никем не оплачено. В крупных государственных структурах все более-менее пристойно, контрафактного ПО там немного. Но чем дальше от Москвы, тем ситуация сложнее. Муниципалитетов больше 20 тыс. в стране. Как правило, там используется много контрафактного ПО, поскольку это бедные организации»<sup>13</sup>.

А если добавить к муниципалитетам школы – что-то около 60-ти тысяч... Представляете, такая констатация обыденности. А мы тут в ВТО намеренно вступили... Не уйти ли после этого в отставку всем, кто призван присматривать за соблюдением в стране законов в отношении ПО? Я только с одним не соглашусь, про «бедные организации». Ведь вот стремятся же они с таким вожделием приобрести проприетарное ПО, судя по реакции на не заставившие себя ждать «письма счастья» от известного мирового производителя, не говоря уж о прочих тратах-растратах. На самом деле, и не стремятся. Хотят не высовываться, выжидают. Вот и все их стремления по этой части.

## Альтернатива есть

И вот что удивительно: в этой зиме живут кактусы. Так можно назвать энтузиастов-романтиков, которые с идиотской (в достоевском смысле) настойчивостью устанавливают, изучают, используют свободное ПО. Их очень мало, а в среде учительства – единицы, ведь зима вокруг. Им тяжело, они не такие как все. Мыслят не так, делают не то, что сказано. Вернее, не только то, что сказано. Получается не так, как у всех, а тоже работает. Это и, соответственно, то, что они есть, не может не раздражать. Им приходится отращивать колючки. Приходится беречь силы про запас, не цвести и даже съеживаться. Сибирская зима – долгая вещь, и они привыкли терпеть, подстраиваться и помалкивать. Они вообще не любят вступать в споры типа «Linux против Windows» (те, кто разжигает такие споры – не кактусы, скорее провокаторы). Но когда их спрашивают, они говорят то, что думают, доказательно и логично, они готовятся к вопросам заранее. Как правило, их ответы – правда, причем на законных основаниях. Конечно, это вызывает консолидированное желание их подавить и выкорчевать, но оно странно сочетается с предчувствием, что они могут понадобиться, что обстоятельства могут сложиться так, что

без них не обойтись. Ведь доходят-таки до «скованных одной цепью»<sup>14</sup> слухи о государственной политике зарубежных краев в области ПО, а также есть подозрение, что там знают толк в деловой выгоде (не только

личной!) и умеют считать. А ну как и здесь понадобится?

Автор этих строк как раз смеет причислять себя к кактусам, причем направленности более идеологической, нежели технологической. Ничего не поделаешь с любовью к независимости и свободе в себе – она необъяснима: она просто есть, или ее нет. Но в другом человеке она может возникнуть тоже, и на это часто направлены помыслы, труды и деяния кактусов. Да, кактусы непременно деятельны и продуктивны. Расскажу об одном таком продукте, поскольку имею к нему непосредственное отношение.

Это сайт для школьного предмета «Информатика и информационные технологии», который авторы называют Интернет-учебником, <http://iiikt.narod.ru/index.htm>. Его особенность в том, что он двухплатформенный: учебный материал в равной степени построен на базе операционных систем Windows и Linux. Последняя – из школьного пакета ПСПО 5 компании «АльтЛинукс».

Учителя слышаны о трудоемкости перехода на свободное ПО, и в большинстве случаев они не готовы к этому переходу не только идейно, но и технологически. Поэтому одна из целей сайта – максимально облегчить переход для тех, кто решился. Ситуация сегодня уже не кажется столь фатальной, как в пору внедрения СПО; при отсутствии основательных универсальных бумажных учебников материала уже достаточно, и сетевая взаимопомощь весьма активна. Но на перевод собственных наработок в другие форматы требуется немало времени и сил (и тут замечу: у начинающих учителей есть преимущество – ничего не надо переделывать). О способности же и желании оказывать помощь со стороны административных и методических структур, не столько техническую (хотя и это – важная задача), сколько именно консультационную, сказано выше. На сегодняшний день в Федеральном перечне<sup>15</sup> только два учебно-методических комплекса пытаются решить эту проблему и декларируют мультисистемность практикума<sup>16</sup>. Кроме того, есть более практичный, но не «грифованный» бумажно-дисковый УМК по информатике Л. Ф. Соловьевой<sup>17</sup>.

В нашем Интернет-учебнике все практические работы сделаны в расчете как для распространенного проприетарного ПО, так и для свободно распространяемого. Причем, в части офисных

## ПРО ПОСТАНОВЛЕНИЯ О СВОБОДНОМ ПО «В отделах образования, а тем более в школах, ничего об этом не знают.»

»



► Кактусы всех образовательных учреждений, соединитесь! Платформа для этого имеется.

технологий они рассчитаны и на входящий в состав ПСПО 5 *OpenOffice.org*, и на вариацию *LibreOffice* для Windows. Форматы практических работ универсальны – PDF или RTF. В очном режиме использования практические работы рекомендуется распечатывать для каждого рабочего места, чтобы поддержать индивидуальную скорость выполнения.

Теоретического материала на страницах Интернет-учебника немного, авторы ограничились самым необходимым на их взгляд, предоставляя учащимся широкую возможность обращаться к соответствующим параграфам утвержденных учебников и доступным цифровым образовательным ресурсам. На сегодняшний день полностью открыт базовый курс предмета для основной школы (8-9 класс) в соответствии с установленной учебной программой 2004 г., дополненный большим объемом элективного материала, что позволяет широко использовать сайт и для старшей школы.

Понятно, что по существующим реалиям и формальным требованиям Интернет-учебник не может стать проводником школьного курса, и, следовательно, годится только как дополнительное пособие. Во всяком случае, следует заявлять его использование именно так, отписывая классные журналы как положено. Но оперативность исправления ошибок и внесения изменений, бесплатность и дистанционная доступность, в том числе и при подготовке к ЕГЭ, делают его достаточно привлекательным по отношению к гораздо более неповоротливым бумажным собратьям. А уж что касается СПО –

теперь довод недоброжелателей об учебной необеспеченности потерял всякую почву и может согреть только совершенно несведущих и до сих пор не подозревающих о существовании глобальной Сети. За то голосует и счетчик посещений сайта – конечно, не как в каком-нибудь Интернет-магазине, но стабильный (надо полагать, учительский) приход им показывается.

Интернет-учебник по информатике запущен в Сеть в августе 2009 г. частным образом исключительно на основе энтузиазма его авторов, и с тех пор постоянно поддерживается и обновляется, неизменно уклоняясь от попыток приручения и подстрегивания внешними силами. В марте 2011 г. он был очно представлен на первом Всероссийском съезде учителей информатики в МГУ<sup>18</sup>. В декабре того же года получил сразу две награды: стал лауреатом всероссийского конкурса сайтов «Позитивный контент-2011» в спецноминации «Лучший сайт педагога»<sup>19</sup> и победителем (1-е место) всероссийского конкурса «ИТ-Прорыв» в категории «IT-обучение» (школьные учителя), номинация «Электронные учебные пособия»<sup>20</sup>.

## А напоследок я скажу...

Пожалуй, с сибирской зимой и кактусами как явлениями текущего момента в школьном образовании все ясно. Картина мерзлая, но ее надо видеть. Иначе не возникнет попыток согреться, не захочется предпринимать действий для оттаивания. И еще я читал, что слова «стужа» и «стыд» имеют один древний корень. Хотелось бы, чтобы эта связь продолжалась.

Ну, а что же борьба? Как и в чем она проявляется? Тут подошли бы слова аббата Фариа<sup>21</sup>, пафосно и эффектно произнесенные

замечательным актером А. Петренко в фильме «Узник замка Иф» (1988): «Сын мой! Зло победить невозможно, потому что борьба с ним и есть жизнь».

## ПРО НАЛИЧИЕ УЧЕБНЫХ ПОСОБИЙ

# «Интернет-учебник по информатике запущен в Сеть и поддерживается.»

Но думаю, что само существование кактусов, их неизменное произрастание, их положительная результативность, их настрой не на обязательность понимания и признания, а на веру в пользу своего дела – и есть их борьба. Им не нужно и некогда культивировать ее самим, борьба возникает вокруг как следствие, как естественная сиоминутная реакция. Им успеть бы выполнить намеченное. Потом люди разберутся. Как скоро – это проблема не кактусов. Остальным остается пожелать хотя бы не мешать им, если уж не умеют их беречь в этой зиме. Таков мой сказ. LXF

## Не будем голословны

<sup>1</sup> <http://www.altlinux.ru/products/5th-platform/school-box/>  
<sup>2</sup> <http://linux.armd.ru/ru/distros/>, <http://www.spohelp.ru/>  
<sup>3</sup> «Информика 6.0 Школьный», <http://www.altlinux.ru/products/6th-platform/informika-school/>; EduMandriva 2011, <http://edumandriva.ru/LinuxWizard>, <http://school.linuxwizard.ru/>  
<sup>4</sup> <http://www.info-altai.ru/news/vpervye-v-rossii-sostoitsya-mezhdunarodnaya-konfer>  
<sup>5</sup> <http://kurs.mioo.ru/reg/courlist-n12.asp>  
<sup>6</sup> <http://www.raspo.ru/blog.html?id=116>  
<sup>7</sup> <http://www.raspo.ru/blog.html?id=119>  
<sup>8</sup> «Сидя на красивом холме» (альбом «День серебра», 1984 г.), [http://www.aquarium.ru/discography/den\\_serebr216.html](http://www.aquarium.ru/discography/den_serebr216.html)  
<sup>9</sup> <http://government.ru/gov/results/13617/>  
<sup>10</sup> <http://www.rg.ru/2012/04/25/internet-dok.html>  
<sup>11</sup> <http://protect.gost.ru/document.aspx?control=7&id=177075>  
<sup>12</sup> [http://www.linuxformat.ru/download/foss-russia/gost\\_54593-2011.pdf](http://www.linuxformat.ru/download/foss-russia/gost_54593-2011.pdf)  
<sup>13</sup> <http://www.raspo.ru/blog.html?id=118>  
<sup>14</sup> Название песни из репертуара группы «Наутилус Помпилиус». Автор стихов – И. Кормильцев  
<sup>15</sup> Федеральный перечень рекомендованных и допущенных учебников на 2012–2013 учебный год, утвержденный приказом Минобрнауки России от 27.12.2011 № 2885, [http://www.edu.ru/db/mo/Data/d\\_11/m2885.html](http://www.edu.ru/db/mo/Data/d_11/m2885.html)  
<sup>16</sup> УМК Н. Д. Угриновича и УМК Л. Л. Босовой и Л. Ю. Босовой, см. <http://lbz.ru/books/376/>  
<sup>17</sup> Можно загрузить с сайта <http://www.bhv.ru/peds/inprint.php>  
<sup>18</sup> <http://it.teacher.msu.ru/tables>, см.: Выграненко М. В., Выграненко М. А. Свободное методическое обеспечение в поддержку перехода на свободное программное обеспечение при обучении информатике в ИКТ. // Всероссийский съезд учителей информатики. Москва, МГУ имени М. В. Ломоносова. 24–26 марта 2011: Тезисы докладов. – М: Издательство Московского университета, 2011. С. 698, 699  
<sup>19</sup> <http://positivecontent.ru/node/1378>  
<sup>20</sup> [http://tvoystart.ru/news/news-41717/?ELEMENT\\_ID=41717&PAGEN\\_1=1](http://tvoystart.ru/news/news-41717/?ELEMENT_ID=41717&PAGEN_1=1)  
<sup>21</sup> Персонаж романа А. Дюма «Граф Монте-Кристо».

# Нордавинд Северо-Запад

## Охранные системы на базе GNU/Linux

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

## Наши услуги и продукты

- Проведение НИР
- Разработка и изготовление систем охранного телевидения и сопутствующих компонентов
- Проектирование сложных систем и комплексов
- Широкий ассортимент готовой продукции

## Наши системы видеонаблюдения предназначены для:

Паркингов

Зданий

Бизнес-центров

Торговых комплексов и складов

Открытых площадок

Жилых домов

Объектов повышенной секретности

Режимных предприятий

## Сертификаты ФСТЭК и ФСБ

4 канальная система видеонаблюдения стоимостью **40 тыс. руб.** —  
**в 2 раза дешевле**, чем предыдущие технологии.



### ТелеВизард

Многофункциональный сервер общего назначения.



### ТелеВизард Авто

Самая бюджетная и высокоэффективная система распознавания автомобильных номеров.



### Источники видеосигнала

Комплекс eyeSense, в состав которого входят видеочамера и объектив. Дополнительное оборудование — ИК-прожектор и климат-защита.

«Нордавинд Северо-Запад»

Научно-производственная компания

Тел. +7 (812)

**309 0686**

Тел./факс +7 (812)

**640 4990**

<http://spb.nordavind.ru>

e-mail: [spb@nordavind.ru](mailto:spb@nordavind.ru)

Приглашаем к партнерству дилеров и проектировщиков систем



Ubuntu – это новая Apple?  
Потерпел ли Linux крах  
на рабочем столе?

Командная строка изжила  
себя? Что важнее: свобода  
или функциональность?

Пожалуйста,  
выйдите вон.

## В ноябрьском номере

# Привет вам, Линус!

Мы посетили г-на Линуса Торвальдса, когда он сидел дома, ожидая какой-то доставки, и расспросили его про эту штуку Linux. Некоторые его ответы стали для нас неожиданностью...

## OpenStreetMap

Как помочь разукрасить карту людям, которые не взимают мзду за использование их информации. Google бяка!

## 3 языка на 4-х страницах

Чтобы разобраться в грамматике языка, надо выучить другие языки. То же касается и языков программирования.

## О, как ты стала хороша

Нынешние дети не знают, что они родились. Оглянемся на недобрые старые времена и оценим величие Linux сегодня.

Содержание будущих выпусков может меняться – вдруг без Джона, нашего морального компаса, мы собьемся с пути.

# 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 3000 экз.

### РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

#### Главный редактор

Кирилл Степанов [info@linuxformat.ru](mailto:info@linuxformat.ru)

#### Литературный и выпускающий редактор

Елена Толстякова

#### Переводчики

Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова

#### Редактор диска

Кирилл Степанов

#### Верстка, долечатная подготовка

Сергей Рогожников

#### Технический директор

Денис Филиппов

#### Директор по рекламе

Владимир Савельев [advert@linuxformat.ru](mailto:advert@linuxformat.ru)

#### Генеральный директор

Павел Фролов

#### Учредители

Частные лица

#### Издатели

Виктор Федосеев, Павел Фролов

#### Отпечатано в ООО «Ланинь»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4  
Заказ 7069

### РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон [Graham Morrison] [graham\\_morrison@futurenet.com](mailto:graham_morrison@futurenet.com)

Заместитель редактора Эндрю Грегори [Andrew Gregory]

[agregory@futurenet.com](mailto:agregory@futurenet.com)

Штатный автор Джонатан Робертс [Jonathan Roberts]

[jonathan\\_roberts@futurenet.com](mailto:jonathan_roberts@futurenet.com)

Редактор диска Бен Эверард [Ben Everard] [ben\\_everard@futurenet.com](mailto:ben_everard@futurenet.com)

Художественный редактор Эфраин Эрнандес-Мендоса

[Efrain Hernandez-Mendoza] [efrain.hernandez-mendoza@futurenet.com](mailto:efrain.hernandez-mendoza@futurenet.com)

Выпускающий редактор Гэри Уокер [Gary Walker] [gary.walker@futurenet.com](mailto:gary.walker@futurenet.com)

### ПОДГОТОВКА МАТЕРИАЛОВ

Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Бен Эверард [Ben Everard], Джульетта Кемп [Juliet Kemp], Джон Лэйн [John Lane], Джеймс Литтон [James Litton], Грэм Моррисон [Graham Morrison], Адам Оксфорд [Adam Oxford], Джонатан Робертс [Jonathan Roberts], Майк Сондерс [Mike Saunders], Маянк Шарма [Mayank Sharma], Ник Вейч [Nick Veitch], Евгений Балдин, Михаил Выграненко, Артем Зорин, Алексей Федорчук, Игорь Штомпель

Художественные ассистенты Стейси Блэк [Stacey Black], Кейт Мак-Доннелл [Kate McDonnell]

Иллюстрации Шейн Коллиндж [Shane Collinge], Елси Уолтон [Ely Walton Illustrations],

iStockPhoto, Саймон Миддлвек [Simon Middleweek]

Фото Джейсон Каплан [Jason E. Kaplan]

### КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Tel. +44 01225 442244 Email: [linuxformat@futurenet.com](mailto: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](mailto:partner@linuxcenter.ru)

**Авторские права:** Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставить уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вам материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

**Ограничение ответственности:** используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

LINUX – зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

«GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



Mandriva Linux — один из самых популярных дистрибутивов GNU/Linux в мире. Главные преимущества Mandriva — дружелюбный интерфейс, простота настройки, возможность быстрой адаптации пользователей, ранее не знакомых с этой ОС, совместимость с широким спектром программного и аппаратного обеспечения.

## Корпоративные продукты Mandriva Linux

### Mandriva 2010.2 Powerpack

Дистрибутив Mandriva 2010.2 Powerpack включает набор офисных и серверных приложений, и подходит для установки на офисной или домашней рабочей станции и на сервере. Дружелюбный интерфейс, простота настройки Mandriva Powerpack, совместимость с широким спектром аппаратного обеспечения и совместимость с «1С:Предприятие» обеспечивают корпоративным пользователям возможность легкого перехода с Windows на GNU/Linux.

### Mandriva Enterprise Server 5

Mandriva Enterprise Server 5 (MES 5) — это надежный и производительный дистрибутив GNU/Linux для корпоративного сервера. MES 5 поможет вам снизить текущие расходы и упростить инфраструктуру. В MES 5 интегрированы серверные разработки программистов Mandriva, в том числе сервер каталогов пользователей Mandriva Directory Server, а также ведущие свободные серверные приложения, которые помогут вам с минимумом затрат времени и энергии настроить и поддерживать необходимые вам серверы. Срок поддержки дистрибутива — 5 лет.

## Сертифицировано ФСТЭК

Дистрибутивы Mandriva Linux сертифицированы по требованиям ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ, что дает возможность использовать их для обработки конфиденциальной информации в автоматизированных системах класса до 1Г включительно и обработки персональных данных в информационных системах класса до К2 включительно.

- **Mandriva 2008 Spring Powerpack** — дистрибутив для рабочей станции или небольшого сервера.
- **Mandriva Corporate Server 4 Update 3** — дистрибутив для создания корпоративного сервера.
- **Mandriva Flash** — дистрибутив GNU/Linux, загружающийся и работающий прямо с USB-носителя.

## EduMandriva — свободное ПО для образования

- Создано с участием российских преподавателей и методистов.
- Все ПО, необходимое для преподавания информатики.
- Методические материалы.

| Наименование  | Стоимость, руб. |
|---|-----------------|
| <b>Корпоративные продукты Mandriva</b>  |                 |
| Mandriva Linux 2010.2 Powerpack (DVD-Box)   | 1 300           |
| Услуга подписки на Mandriva Enterprise Server 5 на 1 год, базовый уровень (с физическим носителем)  | 13 300          |
| Услуга подписки на Mandriva Enterprise Server 5 на 3 года, базовый уровень (с физическим носителем) | 34 800          |
| <b>Продукты Mandriva для образования</b>  |                 |
| Комплект программного обеспечения Mandriva Linux и EduMandriva для школ                             | 3 500           |
| <b>Сертифицированные ФСТЭК продукты Mandriva</b>  |                 |
| Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 10 рабочих мест                           | 28 500          |
| Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 5 рабочих мест                            | 14 500          |
| Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 1 рабочее место                           | 4 990           |
| Сертифицированный ФСТЭК Mandriva Corporate Server 4.0 Update 3                                      | 10 050          |

С вопросами по продуктам Mandriva обращайтесь в ГНУ/Линуксцентр!

**MANDRIVA УЖЕ ИСПОЛЗУЮТ:**  
 МВД РФ, Минздравсоцразвития РФ,  
 Минфин республики Саха (Якутия),  
 Правительство Московской области,  
 администрация Черниговского района,  
 Приморского края, ОАО «Морион»,  
 сеть магазинов «Компьютер-центр  
 «КЕЙ», группа компаний «ИМАГ»,  
 компания «Азбука мебели»,  
 и многие другие.

Москва  
+7 (499)

**271-49-54**

Санкт-Петербург  
+7 (812)

**309-06-86**

Linux-эксперт для вашего бизнеса. [www.linuxcenter.ru](http://www.linuxcenter.ru)



# ЦЕНЫ В СВОБОДНОМ ПАДЕНИИ

HETZNER DEDICATED  
ROOT SERVER **EX 4**

~~5500~~ руб.

**1900** руб.

УСТАНОВКА

HETZNER DEDICATED  
ROOT SERVER **EX 4S**

~~5500~~ руб.

**2200** руб.

УСТАНОВКА

HETZNER DEDICATED  
ROOT SERVER **EX 6**

~~5500~~ руб.

**2900** руб.

УСТАНОВКА

HETZNER DEDICATED  
ROOT SERVER **EX 5**

~~1900~~ руб.

**0** руб.

УСТАНОВКА

## HETZNER DEDICATED ROOT SERVER **EX 4**

- Intel®Core™ i7-2600 Quadcore с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM
- 2 x 3 ТБ SATA 6 Гбит/с 7200 об/м! (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик\*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 1900 рублей

**1900**  
рублей в месяц

## HETZNER DEDICATED ROOT SERVER **EX 5**

- Intel®Core™ i7-920 Quadcore с поддержкой технологии Hyper-Threading
- 24 ГБ DDR3 RAM
- 2 x 750 ГБ SATA 3 Гбит/с (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик\*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 0 рублей

**2500**  
рублей в месяц



GreenIT  
Best Practice Award

2011

Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выберите более чистое будущее вместе с Hetzner Online!



**RU.HETZNER.COM**

\*Трафик предоставляется бесплатно. При превышении порога 10000 ГБ/месяц скорость соединения будет ограничена до 10 MBit/s. Стоимость аренды постоянного канала с пропускной способностью 100 MBit/s составляет 290 рублей за каждый дополнительный ТВ.