

СУПЕР-БЛЕСТЯЩИЙ DVD
Tails: супер-приватность в стиле Debian
PC-BSD, Haiku, Icaros и более того!

LINUX FORMAT

Главное в мире Linux

Июнь 2012 № 6 (158)



Освежите свою ОС

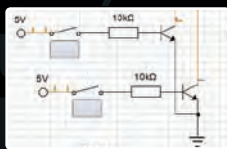
» Пять лучших после Linux операционных систем с. 28

Надуем



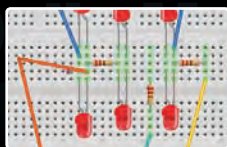
Отгоните злодеев подальше с помощью нашего руководства

Обезопасим web-серфинг » Прибьем cookie » Утаим IP-адрес » Сохраним анонимность



Строим виртуальный CPU

» Сами сделаем ПК, чтобы разобраться в его работе с. 50



Arduino

» Сладкозвучная мелодия платы — шумелка назло соседям с. 78



Киаран О'Риордан

« Ситуация абсурдная... и на суды рассчитывать не придется »

Патентное безумие в сердце Европы с. 42

Плюс!
MINIX
был дедушкой Linux, а теперь рвется в его внуки с. 54

Также в номере...

Нефть — наше все

» Не скупаем в панике бензин, а бомбим нефтяные поля с. 12



Чертим карту

» Ваш маршрут — находка для шпиона с. 44



Android против iOS

» Тест телефона или флирт с девушками? с. 48



Текстовый редактор

Продвигаем Vim

» Пишите коды быстрее и спасите этот мир

Чат онлайн

Осваиваем IRC

» Чтобы срочно придти на помощь FYI AAMOF LXF PPL

Чисто символически

Рисуем шрифт

» Не надпись на гробах... увековечим свой почерк

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

Linux center
www.linuxcenter.ru

Red Hat Enterprise Linux

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

Самая популярная в мире
Linux платформа для бизнеса

Обеспечивает высокую
производительность, надежность,
масштабируемость и безопасность

Сертифицирована ведущими
производителями оборудования
и разработчиками ПО



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

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

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



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

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

Red Hat — ведущий серверный дистрибутив Linux

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

Специальное предложение для читателей Linux Format!

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



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

Что мы делаем

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


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

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


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

Кто мы

Тема этого номера — как сохранить свои данные в секрете от любопытствующих органов власти. Итак, что же вам хотелось бы утаить от Большого Дяди?



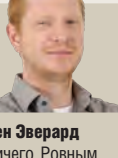
Гэри Уокер
Ш-шш! За мной гонится полиция благоразумия. Свобода — это рабство! Незнание — сила! Мы МЕРТВЫ!



Эндрю Грегори
Не желаю, чтобы кто-то наложил лапы на мои сохраненные игры *Frontier: Elite 2*. Я заработал эти баллы в честном бою!



Эфраин Зрианес-Мендоса
Тайную подписку на незаконное потоковое вещание с www.liveMексика-Футбол2-йЛиги.com.



Бен Эверард
Ничего. Ровным счетом ничего. Я чист, чист, поняли?.. Стойте, это что — стук в дверь? На помощь, моя верная красная таблетка...



Маянк Шарма
Мой пароль сайта LoveFilm. «Возвращение Короля» висит в моем списке заказов восемь лет. Его скоро пришлют, я это чую.



Джонатан Робертс
Планы вторжения корнуэльцев в Девоншир. Сегодня — прибрежные курорты, завтра — весь мир у моих друзей.



Майк Сондерс
Мой проект взломать компьютеры ведущих мировых финансовых учреждений и посадить их всех на MikeOS.



Валентин Синецын
Свою подпись Linux Format. Между страницами я храню записки и коллекцию вкладышей от жвачек.



Ник Вейч
Мою богатую коллекцию пиратски скачанных MP3 Кристины Агилеры с ее подло недооцененным новым альбомом.



Сюзан Линтон
Космолет — я тайно монтирую его в гараже. Думаю покинуть эту обреченную Землю и открыть в 2020 г. планету Линтон.



Шашанк Шарма
Большой адронный коллайдер — последние 30 лет я строил его под кустом розмарина в моем саду. Он мой, только мой!



Нейл Боткин
Оффшорные миллионы, нажитые писанием Ответов для LXF за последние 85 лет. Это же не облагается налогом, а?

Знание — сила

» Причем сила враждебная. Похоже, что именно этой логике следовало руководство крупнейшего в США книжного магазина Barnes&Noble, запретив продажу **LXF154**, где темой номера было изучение атак и взломов. Описание работы с *Back Track* они сочли потенциально опасным.

Странно, но при этом на сайте компании доступны электронные, для чтения на NOOK, версии *A Guide to Kernel Exploitation: Attacking the Core, SQL Injection Attacks and Defense* и другие книги издательства Elsevier Inc. («на бумаге» их издает Syngress), которое уже не первый год выпускает серию книг о различных аспектах информационной безопасности. Те из них, что я успел прочитать, в большинстве своем оказались очень хороши. Могу рекомендовать всем интересующимся. Жаль, что русских переводов книг этой серии пока маловато.

А вот в «бумажном» виде Barnes & Noble может предложить кое-что «погорячее», например *Middle Eastern Terrorist Bomb Designs* и другие книги издательства Paladin Press со столь же занимательными названиями. Иногда не новье, но, по утверждению магазина, в очень приличном состоянии.

Кстати, по той же логике должен быть запрещен и этот номер. Статья о сохранении приватности в Интернет несомненно может поспособствовать международному терроризму. Посмотрим. Следите за новостями.

Кирилл Степанов
Главный редактор
» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Вопросы распространения: sales@linuxformat.ru
Web-сайт: www.linuxformat.ru

» Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15
» Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112

Содержание

Иметь секреты вполне естественно – не все стоит выставлять на публику.

Обзоры

Oil Rush 12

Частично стратегия, частично стрелялка, частично зловещее пророчество.



➤ Поубивайте людей из-за нефти и найдите выгодную работу в болтологии.

XBMC 11 Eden 14

Превратите свою Linux-машину (или Raspberry Pi) в многофункциональный медиа-центр.

Gnome 3.4 15

Этот рабочий стол намерен всех задавить – и на пути к своей цели улучшил браузер.

Audacity 2.0 16

Редактирование аудио в свободном ПО обзавелось VAMP и другими акронимами.

Renoise 2.8 17

Эта цифровая рабочая станция для аудио задела нужные струнки в душе автора обзора.



➤ Renoise – отличная штука. Берем!

Ядро Linux 3.3 18

Выход этой версии принес массу долгожданных изменений. А также дружбу с Android.

PocketBook A 10" 20

Отечественная читалка неплохо справляется со своей работой. Да и ценой явно порадует.

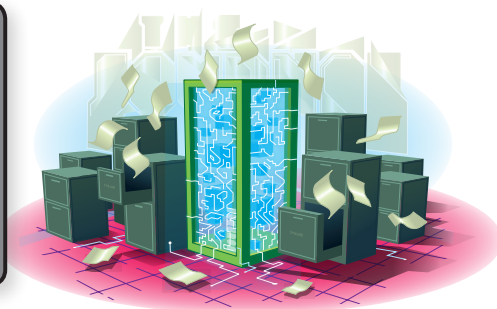
Надуем



А заодно поможем уцелеть
правозащитником с. 34



Сравнение:
Альтернативные ОС с.28



Что за штука –
The Journal? с.58

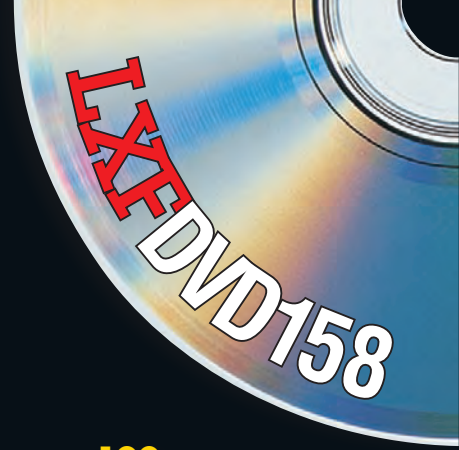
Люди говорят



“Чтобы исключить ПО [из патентования], надо доказать, что это не технология”

Киран О'Риордан ищет резоны в безумии с. 42

На вашем бесплатном DVD



Tails

» Настройка Tor без головной боли

Альтернативные ОС

» Попробуйте нечто непохожее

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

Ищите в этом номере



Ходили вы походами 44

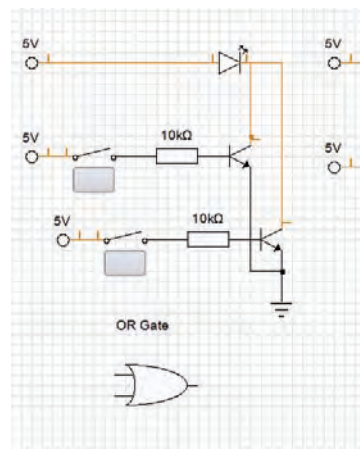
Подкрепите картой несусветные байки о своих приключениях.

Android против iOS 48

Сражение двух мобильных титанов за умы пользователей.

Проектируем CPU 50

Ну чем мы хуже Алана Тьюринга?



Пропустили номер?

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



Учебники

Типографика	
Создаем шрифт	66
Как превратить ваши каракули во вполне приличную гарнитуру.	
Чат онлайн	
IRC	70
Поговорить все любят, а особенно приятно покататься онлайн с незнакомцами.	
Редактирование текста	
Vim	74
Освоим продвинутые функции этого популярного текстового редактора.	
Электроника	
Arduino	78
Ваша плата вдохновенно играет «Оду к радости» – Бетховен утирает слезы умиления...	
Языки программирования	
Erlang	82
Что наша жизнь? Многозадачность! И в Erlang о ней любезно позаботились.	
Безопасность	
OpenVPN	86
Примените VPN, чтобы обеспечить доступ пользователей к сети, а шелупонь отогнать.	
Компьютерная верстка	
LaTeX	88
Автоматизируем задачи набора текстов, генерируя макрокоманды пачками.	

Постоянные рубрики

Новости	4	Что за штука	58	Диск Linux Format	102
Создана первая российская ассоциация по облачным технологиям, готовится открытый стандарт, идет грызня за патенты, а анонимность в Сети нашла защиту.					
Android	22	Рубрика сисадмина	60	Содержимое двустороннего DVD этого месяца.	
Тяжбы за патенты не прекращаются, но программирование идет своим путем.					
Сравнение	28	Ответы	92	Пропустили номер? ...	104
Ищете альтернативную ОС? Считайте, что вы ее нашли.					
Интервью LXF	42	ПРОБЛЕМЫ LINUX РЕШЕНЫ!		Еще не поздно заказать любой из предыдущих выпусков журнала. Жизнь слишком коротка, чтобы тратить ее на тоску о потере.	
Киран О'Риордан – боец невидимого фронта борьбы за свободу ПО.					
Hotpicks	96	Школа LXF	108	Образовательный проект EduMandriva набирает обороты – обзор этапов большого пути.	
Отведаете горяченького: лучшие в мире новинки свободного ПО.					
		Через месяц	112	Ожидание практически нестерпимо, но очередной номер в очередной раз прибудет в срок!	



ГЛАВНОЕ Тревоги открытого стандарта » Tor — страж свободы в Сети
» Русь витает в облаках » Цензура в книжной лавке » Открытый робот

СТАНДАРТ ДЛЯ ВСЕХ ЕДИН

Критерии открытого стандарта



» Рубрику готовил
АРТЕМ ЗОРИН

Совместно с Правительством Великобритании Организация Open Source Initiative разработала правила для всех.

Правительство Великобритании проводит опрос на тему того, какие типы патентного лицензирования необходимы открытым стандартам. Нужда в подобном опросе возникла после поступления предложения разрешить использование в стандартах патентов, распространяемых на условиях FRAND, которые могут требовать выплаты лицензионных отчислений, а также ограничивать разработку, распространение и использование ПО, где реализован подобный стандарт. Для рассмотрения опасностей такого шага и выработки соответствующих рекомендаций Европейский фонд свободного ПО организовал саммит экспертов.

В противовес условиям FRAND ставится требование к держателю патента предоставить каждому свободную от денежных отчислений лицензию при использовании в реализации стандарта запатентованных технологий. Такая мера используется организацией W3C при формировании веб-стандартов, что дает возможность свободному ПО поддерживать такие стандарты. Если стандарт пересекается с патентами, требующими лицензионных отчислений за каждую копию распространяемой про-

граммы, его поддержку невозможно реализовать в свободном ПО, так как необходимость сбора отчислений за каждую копию программы нарушает основной принцип свободного ПО – возможность неограниченного распространения.

«Сбор отчислений нарушает принципы свободного ПО.»

Тем не менее, существование патентов на условиях FRAND не исключает свободное ПО автоматически. Некоторые из патентодержателей предоставляют либеральные патентные лицензии на любое использование в конкретной программе, и иногда эти лицензии покрывают любую программу, распространяемую на условиях GNU GPL. Но подобные случаи являются редким исключением, только демонстрирующими пропасть, которая зияет между типичными FRAND-лицензиями и свободным ПО.

Саймон Фиппс [Simon Fipps], член управляющего совета организации Open Source

Initiative (OSI), сформулировал список критериев, которым должен соответствовать открытый стандарт. Критерии разработаны с учетом требования, что «открытый стандарт» не должен мешать созданию реализации стандарта в открытом ПО.

Критерии открытого стандарта:

- » **Отсутствие преднамеренных секретов** Стандарт не должен утаивать ни единой подробности, необходимой для его переносимой реализации.
- » **Доступность** Стандарт должен быть публично доступным (размещенным на стабильном сайте), на условиях, свободных от выплаты отчислений. Сам текст стандарта может продаваться по разумной цене без дискриминации.
- » **Патенты** Все патенты, необходимые для реализации стандарта, должны удовлетворять одному из двух пунктов:
 - » быть лицензированы на условиях, исключающих денежные отчисления и без ограничений на область использования;
 - » или должны покрываться договорной обязанностью по отказу от правопритязаний в случае использования в открытом ПО.
- » **Отсутствие соглашений** Должны отсутствовать любые соглашения (лицензионные, соглашение о неразглашении, гранты, рекламные и любые другие формы бумажных документов) для возможности неограниченного размещения и внедрения реализации, соответствующей стандарту.
- » **Отсутствие зависимостей, несовместимых с критериями открытого стандарта** Реализация стандарта не должна требовать другой технологии, не соответствующей данным критериям.

FRAND нам не товарищ

FRAND – Разумное и Не Дискриминационное лицензирование (РиНДЛ) (англ. Reasonable and Non-Discriminatory, RAND) – термин, использующийся при процессах стандартизации.

В общем случае это означает, что присоединяющиеся к процессу стандартизации компании соглашаются: если они получают какие-нибудь патенты на тех-

нологии, входящие в стандарт, то разрешают другим использовать эти патенты на равных условиях, и если ограничения будут, то только разумные.

Такое лицензирование позволяет брать с пользователей патентов «разумную» плату за каждого покупателя, и поэтому может быть несовместимо с понятием свободы ПО.

1С:ПРЕДПРИЯТИЕ 8

Новое поколение систем автоматизации управления и учета

- 1С:Управление производственным предприятием
- 1С:Бухгалтерия
- 1С:Зарплата и управление персоналом
- 1С:Управление торговлей
- Отраслевые и специализированные решения

Теперь работает в Linux!

Для запуска «1С:Предприятия» в Linux необходимо наличие WINE@Etersoft, который входит в состав дистрибутива Mandriva Linux 2011 Powerpack



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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



OPEN SOURCE ПРОТИВ ТИРАНИИ

Инструмент сетевого мониторинга

Разработчики Tor намереваются создать глобальную карту ограничений свободы в Интернете.

Разработчики системы обеспечения анонимных сетевых соединений Tor – Артуро Филасто [Arturo Filasto] и Якоб Аппельбаум [Jacob Appelbaum], объявили о создании сети OONI (Open Observatory of Network Interference) для наблюдения за сетевыми вмешательствами и разработке инструмента OONI-probe для сетевого мониторинга. Проект предназначен для автоматического обнаружения и последующего документирования применения любой формы скрытой сетевой цензуры в Интернете по всему миру. Инструмент OONI-probe дает пользователю возможность проверить свое интернет-соединение на наличие факта цензуры и ограничений доступа к определенным сайтам, а также провести тестирование канала связи на предмет выборочного ограничения полосы пропускания.

Проект OONI частично финансируется за счет гранта от Radio Free Asia. Согласно утверждениям одного из разработчиков, главная цель проекта OONI – создание открытого фреймворка для контроля за сетевой цензурой, изначально спроектированного таким образом, чтобы любой исследователь мог независимо проверить методологию обнаружения факта цензуры, а также надежно воспроизвести эти тесты в своем конкретном случае. Ключевой мотив проекта – создание возможности свободного обмена данными и вовлечение в накопление данных энтузиастов, так как существующие аналогичные средства мониторинга, такие как Google Transparency Report, не афишируют методы сбора данных и не являются прозрачными для анализа. Программа уже активно используется различными политическими активистами для сбора документальных доказательств блокирования

различных политически неугодных сайтов на уровне провайдеров отдельных государств (так, недавно установлен факт скрытой цензуры в сетях Палестины).

Утилита OONI-probe проверяет доступность сайтов по списку, сформированному из миллиона наиболее популярных сайтов мира по версии Alexa в данном регионе исследования, на что уходит, как правило, около двух недель сканирования. Также предусмотрена возможность создания централизованно координируемой сети из группы машин в разных частях света для проведения непрерывной совместной проверки. Любой, кто устанавливает и использует этот инструмент для исследований своего сегмента сети, выступает в качестве волонтера и высылает все полученные технические данные на центральный сайт проекта OONI, где они агрегируются и обрабатываются. Авторы уверены, что когда этот проект получит достаточное географическое распространение, можно будет достоверно увидеть, как выглядит

«Проект предназначен для обнаружения любой формы цензуры.»

глобальный Интернет с самых разных его уголков по всему миру, а также понять, где и какие сайты блокируются или как-то модифицируются.

Инструментарий OONI-probe написан на языке Python, его работа базируется на трех основных концепциях: Наборы [Assets], Тесты [Tests] и Отчеты [Reports]. В Наборах перечисляются коллекции любых сетевых перечисляемых объектов (IP-адреса, URL, ключевые слова и т.д.),



► Один из создателей OONI и TOR — Якоб Аппельбаум.

которые передаются в следующий блок обработки – Тесты. В тестах – логической основе этого комплекса – реализованы наборы правил, которые обрабатывают переданные им данные из указанного Набора. Каждый тест разделяется на экспериментальную и контролирующую часть; их количество может быть произвольным, а содержимое очень разноплановым, они постоянно дорабатываются и добавляются по мере развития комплекса. Отчеты – это стандартизированный набор данных полученных в результате работы тестов; кроме того, данная часть отвечает за загрузку данных на центральный сервер проекта.

Планируется создание четвертого логического блока – постпроцессинга, который позволял бы гибко обрабатывать полученные в результате тестов данные, например, для визуализации результатов каждого отдельного типа тестов, или локально выделять из сырых данных важные логические особенности, вместо отправки сразу всех данных на центральный сервер. На данный момент, несмотря на то, что инструмент уже работает и может быть использован любым желающим, он находится в стадии активной разработки – в частности, к нему отсутствует графический интерфейс, который будет создан позже.

Напомним, что проект OONI – это уже второе, более общее исследование разработчиков системы Tor, направленное на получение данных по цензуре в масштабах всей сети, тогда как в первом своем исследовании они смогли показать попытки активного блокирования работы системы Tor со стороны отдельных государств, а также попытки спецслужб встраивать в систему Tor собственные узлы, в целях контроля и прослушивания трафика.

Серьезная проблема

Интернет-цензура – контроль и пресечение публикации или доступа к информации в сети Интернет. Своим появлением интернет-цензура обязана отсутствию национальных границ в Интернете. Общую проблематику интернет-цензуры можно определить так: информация, противоречащая интересам государства (текущего правительства) и заблокированная на внутренних ресурсах, может быть опубликована на web-серверах в других странах. Цензура в Интернете подразумевает

не только блокировку или закрытие web-ресурсов, но и анализ трафика, организацию лже-оппозиционных web-ресурсов и применение карательных мер к авторам публикаций или владельцам web-серверов.

По данным общественной организации «Репортеры без границ», за 2011 год закрыто или заблокировано 1740 сайтов, 105 блоггеров пострадали за публикацию материалов в сети Интернет, из них 1 – убит, 59 – арестованы, 45 – подверглись физическому нападению.

softline®

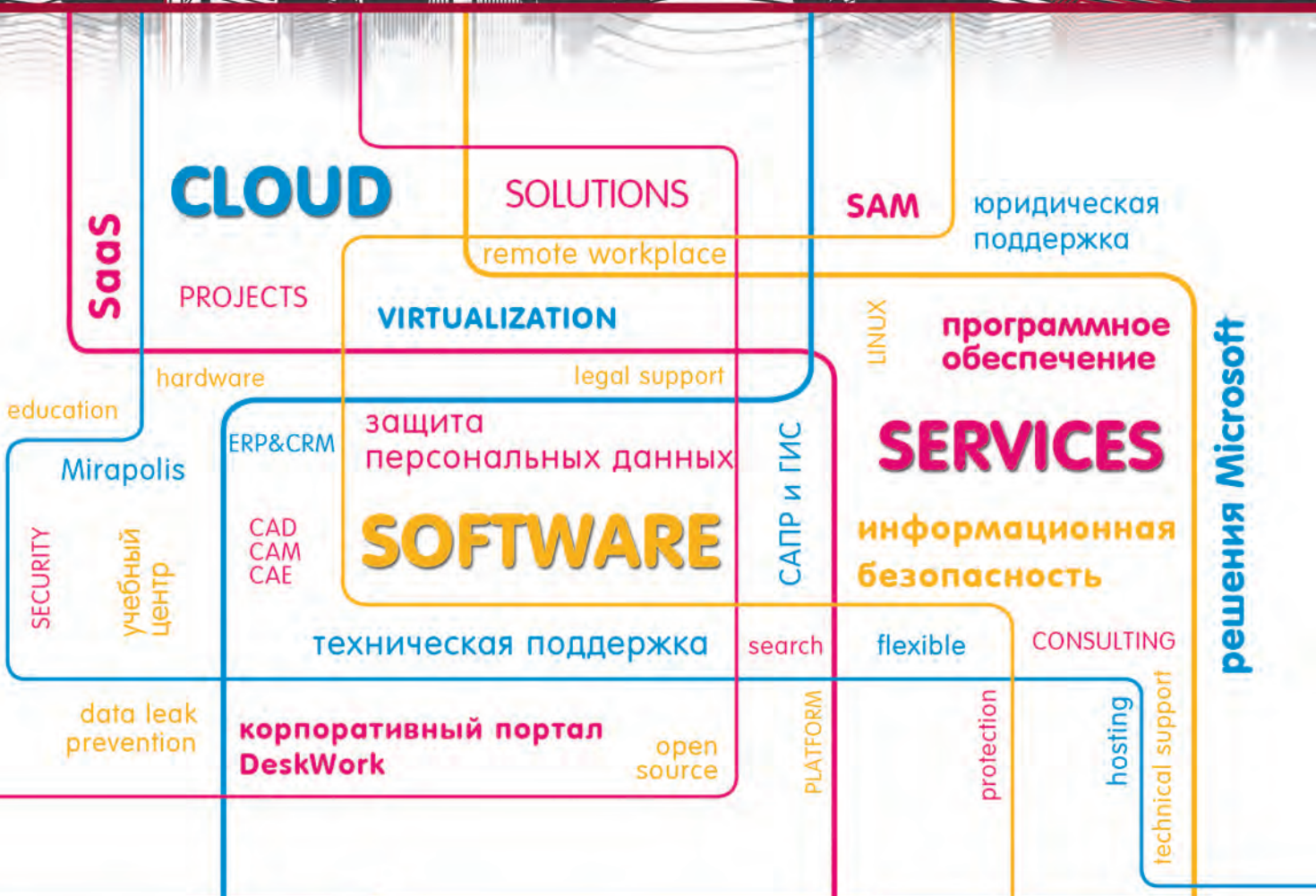


Services

Software

Cloud

ИТ-архитектура вашего бизнеса



+7 (495) 232-00-23

www.softline.ru

info@softline.ru

ДЛЯ ПРОФЕССИОНАЛОВ

Российский облачный союз

В России создана первая профессиональная ассоциация в сфере облачных технологий.

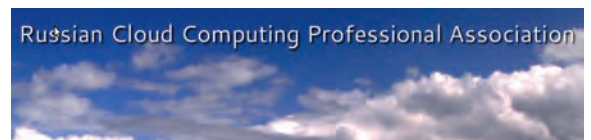
Москва, 26 апреля 2012 г. Группа профессионалов в сфере информационных и облачных технологий объявила о создании профессиональной ассоциации в сфере облачных технологий – Russian Cloud Computing Professional Association (RCCPA). К ассоциации присоединились ведущие эксперты компаний EMC, Microsoft, Softline и многие другие. Вступление в ассоциацию бесплатно, и сделать это можно после регистрации на сайте RCCPA (www.rccpa.ru).

Russian Cloud Computing Professional Association – первое профессиональное объединение независимых экспертов, работающих в области создания, развития и внедрения облачных технологий и сервисов. Ассоциация создана в апреле 2012 года группой активистов облачного рынка.

Основная цель RCCPA – выработка единых подходов к формированию развития облачных вычислений в России и формирование экспертной площадки для развития российских облачных проектов на международных рынках.

Оргкомитет планирует развивать и стандартизировать облачные технологии в России, а также приносить лучшие западные практики и решения на рынок. В планах ассоциации – регулярное проведение встреч, семинаров и конференций, для обмена опытом по профильным вопросам.

Члены ассоциации впервые официально собрались на конференции CloudConf 2012. Чтобы принять участие, было необходимо зарегистрироваться на www.cloudconf.ru.



➤ Новая организация призвана помочь развитию облачных технологий в нашей стране.

Оргкомитет RCCPA собирается проводить ежемесячные события, посвященные развитию рынка облачных технологий. Следить за мероприятиями можно на сайте www.rccpa.ru.

Что такое CloudConf

CloudConf – профессиональная конференция для российских разработчиков, провайдеров и пользователей облачных вычислений, самое крупное и значимое мероприятие по облачным технологиям в России.

В рамках CloudConf по решению жюри, сформированному из представителей интеграторов, вендоров, провайдеров, телекоммуникационных компаний и инвесторов, вручается награда Cloud Award.

Программа конференции CloudConf 2012 включила выступления более 50 докладчиков, среди

которых были представители крупнейших мировых и российских корпораций, занимающихся облачными технологиями, таких как Microsoft, Google, Parallels, VMware, J'son & Partners, Selectel, Tekmi, 1С-Битрикс, МойСклад, IToolabs и многие другие. Организатором мероприятия выступила компания Softline.

Церемония вручения награды на CloudConf 2012 организовывалась впервые. В дальнейшем планируется проводить ее на конференциях CloudConf регулярно, раз в год.

ИХ БЫТ, ИХ НРАВЫ

Искусство взлома забанили

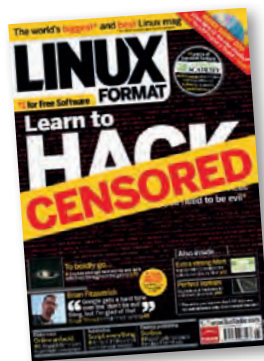
Barnes & Noble, один из ведущих мировых реселлеров электронной периодики и книг, убоился *Linux Format*.

Англоязычный выпуск LXF154 подвергся остракизму в компании Barnes & Noble. Журнал был посвящен «искусству взлома» компьютеров под Linux. По мнению компании, читатели могли бы использовать материалы, изложенные в журнале, не только для самообразования, но и в буквальном смысле – для взлома серверов.

LXF154 был изъят из магазинов Barnes & Noble на всей территории США после «анонимной жалобы».

Однако содержимое той самой статьи все же дошло до своего читателя: наши английские коллеги выложили его в Сеть. Владеющие английским могут прочесть материал по ссылке: <http://www.tuxradar.com/content/learn-hack>.

Истинной целью номера LXF154 в целом и статьи «Изучите взлом» в частности



являлось развенчание мифов о технологиях безопасности, доведение до «простых смертных» информации о современных системах безопасности серверов на основе Linux, а также иллюстрация того, насколько они на самом деле могут быть уязвимы для атак.

➤ Цензура добралась и до англоязычного *Linux Format*...

Так или иначе, для жителей США англоязычный *Linux Format* по-прежнему остается доступен в магазинах приложений Apple, Google, а также в Центре приложений Ubuntu – не все книготорговцы принимают инсинуации анонимных заявителей всерьез...

Про Barnes & Noble

Barnes & Noble, Inc. – старейшая американская компания, крупнейшая в США по продажам книг, один из мировых лидеров по обороту продаваемых книг и услуг через Интернет и один из первых интернет-сервисов, ориентированных на продажу реальных товаров массового потребления. Штаб-квартира расположена в Нью-Йорке (штат Нью-Йорк). Компания стала знаменита в среде OpenSource после продолжающегося до сих пор скандала с Microsoft, который разгорелся из-за патентов.

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

ОТКРЫТЫЕ РОБОТЫ — НАШИ ДРУЗЬЯ

Американцы заботятся о роботах

Представлен фонд поддержки развития открытых решений для робототехники (OSRF).

Американская компания Willow Garage учредила специализированную некоммерческую организацию Open Source Robotics Foundation (OSRF), главная миссия которой — поддержка разработки, распространения и адаптации программного обеспечения с открытыми исходными кодами для его использования в области робототехники, а также для исследований, разработки, популяризации и образования в этой области.

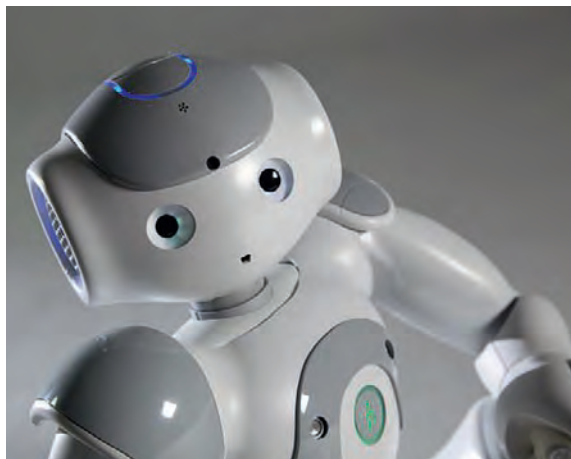
Напомним, что несколькими годами ранее Willow Garage создала свою собственную операционную систему Robot Operating System (ROS), поставляемую под лицензией BSD, которая уже оказала заметное влияние на развитие робототехники. Последняя версия ROS под кодовым названием “Fuerte Turtle” была выпущена в прошлом месяце.

Второй известный проект компании — Gazebo, трехмерный динамический программный симулятор функционирования роботов, поддерживающий имитацию сразу множества роботов в естественной окружающей 3D-среде. Сюда входит как внешний рендеринг их движений и перемещений в заданном интерьере [rigid-body physics], так и одновременная имитация внутреннего функционирования всей программно-аппаратной начинки робота, включая мониторинг и отладку сервисных сообщений между всеми узлами робота при его движениях и работе в режиме реального времени. Gazebo поддерживает такие ОС, как ROS и Player.

Оба эти проекта уже переданы для курирования их развития под начало Open Source Robotics Foundation.

Будучи формально основанным в марте 2012 года, фонд OSRF только сейчас публично представил свой окончательный управляющий комитет. Так, в состав текущих управляющих фондом включены:

профессор Вольфрам Бунгард [Wolfram Bungard], который возглавляет Лабораторию автономных мыслительных систем [Laboratory for Autonomous Intelligent Systems] в университете города Фрайбург; Райан Гэрипи [Ryan Gariepy] — один из основателей и главный технический ди-



ректор компании Clearpath Robotics; Брайан Джерки [Brian Gerky] — директор Open Source Development в компании-основателе фонда Willow Garage, Хелен Грейнер [Helen Greiner] — одна из основателей iRobot и текущий руководитель компании CyPhyWorks; Сэм Парк [Sam Park] — представитель южнокорейской робототехнической компании Yujin Robot.

► **Aldebaran Robotics Nao — открытый робот-гуманоид.**

«Создание фонда OSRF — логический шаг для развития отрасли.»

Willow Garage отмечает, что после отдельных удачных проектов в области робототехники было принято решение по созданию фонда OSRF, как следующий логический шаг для координации развития целой отрасли робототехники: для совместной выработки и согласования всех составляющих процесса разработки роботов сообществом единомышленников, изначально основанного на принципах открытых технологий.

Конечная цель развития — не только сугубо академические и научные задачи: преследуются также и практические цели, например, создание гуманоидного типа роботов, которые смогут эффективно помогать человеку как в области медицины, так и в разнообразных спасательных операциях. **LXF**

Новости короткой строкой

» 12 мая в Омске прошла Битва дистрибутивов, а также GNU/Linux Install Fest. Победил дистрибутив Ubuntu Linux.

Источник: <http://omsklug.com>

» 16 мая с 13:00 по 19:00 в Московском энергетическом институте прошел День свободного программного и аппаратного обеспечения с участием представителей ГНУ/Линуксцентра.

Источник: www.linuxformat.ru

» На состоявшемся 29 апреля заседании акционеров французской компании Mandriva удалось достигнуть согласия в вопросе проведения рекапитализации в течение 10 дней.

Источник: <http://blog.mandriva.com>

» Компания Canonical анонсировала первую стабильную версию специализированной сборки для корпоративного сектора и государственных учреждений — Ubuntu Business Desktop Remix, основанную на пакетной базе Ubuntu 12.04 LTS.

Источник: <http://blog.canonical.com>

» Компания Canonical намерена в следующем году довести долю ПК, поставляемых с Ubuntu Linux, до 18 млн экземпляров, что соответствует примерно 5% от всех продаваемых ПК.

Источник: www.phoronix.com

» Высший суд Европейского союза, постановления которого не могут быть обжалованы, вынес решение, что языки программирования, API и возможности компьютерных программ не могут быть отнесены к интеллектуальной собственности и не подпадают под защиту копирайта.

Источник: www.itworld.com

» Компания Motorola Mobility добилась блокирования в Германии продаж Xbox и Windows 7, в которых используются технологии кодирования видеокodeком H.264.

Источник: www.bbc.com

» Разработан открытый драйвер для сенсорных панелей Microsoft Surface 2.0. Драйвер функционирует на пользовательском уровне и использует для взаимодействия с панелью библиотеку *libusb*. Код драйвера и демонстрационное приложение опубликованы на GitHub.

Источник: <https://lists.launchpad.net>



АЛЕКСЕЙ ФЕДОРЧУК
...простой советский юзер.

openSUSE: И на солнце есть пятна

Прошлую колонку я закончил на том, что openSUSE мог бы претендовать на звание оптимального настольного дистрибутива. Что же не дает ему взойти на «пьедестал почета»? Всего две мелочи.

Во-первых, среди многочисленных достоинств openSUSE нету визуального быстрого действия, столь выделяющего, например, Fedora. И можно сколько угодно рассуждать о том, что оно не влияет на скорость выполнения реальных задач – зато оно определяет первое впечатление пользователя, да и позднее – психологический комфорт в дистрибутиве.

Во-вторых, структура репозитория openSUSE не проста. Почти любой пакет вне официально поддерживаемых требует подключения своего репозитория. Это относится и к мультимедиа-приложениям с кодеками, и к «улучшенным» шрифтам, и к играм, и даже к «фронтальным» версиям ряда основных компонентов.

Первое «пятно» отчищается с помощью prelink. Скажем, в Fedora prelink давно принят по умолчанию. Применение prelink в openSUSE значительно снижает исходную ее «задумчивость». Впрочем, уже по тестируемому ныне релизу видно, что майнтейнеры всерьез озаботились визуальным быстрым действием.

Второе «пятно» закрасил бы эффективный механизм поиска пакетов и установки их «одним щелчком». Однако отсутствие централизованного хранилища «сторонних» приложений – аналога RPMFusion в Fedora или AUR в Arch Linux – по-прежнему чувствуется. Но, может быть, и эта задача со временем будет решена?
alv@posix.ru

Сегодня мы рассматриваем:

- Oil Rush** 12
«Ну что в войне хорошего?» – распевали в 1970. Сегодня мы знаем, что войны хороши для акционеров крупных нефтяных компаний, для политиков, выбранных с малым перевесом, и для вдохновения игровых разработчиков. Вот одна такая игра, причем развеселая.
- XBMC 11 Eden** 14
Xbox Media Centre способен превратить ваш ПК в настоящий медиа-центр, и если у вас есть кабель (или Raspberry Pi с липучкой), вы сможете воткнуть его в телевизор и власть насмотреться передач про шпионов. Все-таки ЦРУ не даром деньги платят.
- Gnome 3** 15
Помните, как с каждым новым релизом KDE прибывали новые приложения? Теперь очередь Gnome 3. Новинка данного релиза – капитально улучшенный браузер *Epiphany*.
- Audacity 2.0** 16
Аудиофилы любят кромсать звуковые файлы, а потом склеивать их заново, и *Audacity* – идеальная среда для подобных упражнений. Да и имя у нее бодрит.
- Renoise 2.8** 17
А еще аудиофилы любят программы, издающие электронные сигналы, так что наш Грэм в этом месяце славно отвел душу. Или нет? Проверьте это.
- Ядро Linux 3.3** 18
Исходный код ядра зашкалил за 15 миллионов строк – вы не ошибетесь, предположив, что такое количество обязательно перешло в достойное качество.
- PocketBook A 10"** 20
Этот планшет явно заинтересует ограниченных в бюджете любителей модных девайсов: в нем есть все, что полагается, за сравнительно скромную сумму.

Oil Rush с. 12



› Глобальное потепление – жуткая вещь, особенно если вы живете на болоте. К счастью, оно пока не наступило.

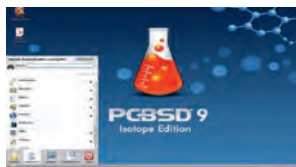
Renoise с. 17



› Где-то за \$80 вы получите на вашем ноутбуке с Linux блестящую студию звукозаписи. Это будет сделка века!

Сравнение: Альтернативные ОС

PC-BSD



GhostBSD



Icaros Desktop



OpenIndiana



Haiku



Haiku, PC-BSD и прочие ОС из этой компании вряд ли в ближайшее время завоюют настольные ПК, но они богаты функциями. Попробуйте их все, и вы увидите, на что способна ваша машина!

Oil Rush 1.07

Алекс Кокс наблюдает за будущим нашей планеты, тратя электроэнергию на стратегию в духе Гринпис.

Вкратце

» Полу-стратегия, полу-защита крепости, но со своей индивидуальностью. См. также *Warzone 2100* или *Tribal Trouble*.

Мир сошел с ума. Моря вышли из берегов, затопив землю. Началась битва за ресурсы. Хорошо еще, что кто-то построил нефтяные вышки, которые теперь разбросаны по Водному шару, в ожидании нового владельца – того, чей флот окажется сильнее. Добро пожаловать в текущий мир *Oil Rush*.

Вам выпало побыть в шкуре австралийского моряка по имени Кевин – то ли это реверанс Кевину Костнеру в *Водном мире*, то ли самое скромное имя для героя [у англичан это имя считается простецким, – прим. ред.]. Подчиняетесь же вы угрюмому парню по имени Командор, отдающему приказы разбить всяких там противников и обеспечить подачу нефти любым способом. Причем вашего согласия никто не спрашивает. Голос этого малоприятного типа – как и большинство голосов в игре – звучит свирепо. Но кампания *Oil Rush*, цель которой ясна с момента, когда вы поступаете в распоряжение этого белобрового маньяка, не дает времени на выбор. Нужно взрывать.

Метаморфозы жанра

В основе своей, *Oil Rush* – смесь стратегии в реальном времени с защитой крепости. Вам нужны ресурсы. Захватывая разные станции, вы получаете известное количество боевых единиц – юнитов. С их помощью вы сможете взрывать вражескую технику. Пока все по законам жанра стра-



» Сражение может стать серьезным испытанием, и вступая в него, нужно знать свои силы.

тегии. На военных базах и баррикадах можно также понастроить себе орудийных башен, насколько ресурсы позволяют. Оттуда же наносятся удары по врагам. А это уже защита крепости.

Но, как всегда в погоне за двумя зайцами, *Oil Rush* не является шедевром ни в одном из этих жанров. Вроде и есть специальные базы, где можно поставить свои

орудия, но управление ими никакой пользы не приносит. Фантазии тоже негде разгуляться – если есть место, а у вас есть нефть, можно создать там боевое укрепление. Блуждающие боевые единицы даже более ограничены: все ваши возможности тонкого управления – «иди туда» да «вернись назад». Никакого перенаправления по ходу, никаких хитроумных маневров.



Шаг за шагом: Ключи к успеху



» Следите за картой

Красные точки – ваши. Если на вашей территории появляются другие, уничтожайте их.



» Защищайтесь

Как вариант, выставляйте у своей базы недорогой, но довольно надежный караул.



» Войте числом

Атакуйте вражеские базы всеми наличными силами одновременно.

Такое впечатление, что пилоты ваших катеров так рьяно давят на газ, что им просто некогда вертеть рулем.

Тем не менее, ресурсы можно распределять – как по типу, так и в процентном соотношении. Например, вы можете бросить в бой авиацию, если вам выпало сражаться на пересеченной местности, или выслать вперед самые шустрые катера, если тяжелые пушки замедляют ход основной фаланги. Также полезно иногда оставлять орудия в тылу: численное превосходство – дело хорошее, но раз защитные башни не способны обезопасить буровые вышки, то пара боевых единиц обеспечит разницу между действующим заводом или его потерей. Делить штатные единицы вы можете только начесть или пополам, но в этом случае простота, пожалуй, и к лучшему.

Когда вы пройдете несколько миссий и насытитесь стандартными трюками обоих жанров, *Oil Rush* швырнет вам крученный мяч: создаст ощущение, что суть игры – исключительно в управлении ресурсами. Возникают препятствия,

«Яркие визуальные впечатления от игры ожидаемы.»

непреодолимые без полного арсенала. Появляются типично стратегические возможности пополнить запасы нефти – дополнительные боевые единицы, более солидные башни и лишние бонусы от дерева технологий – способные доставить вам взятие башни либо спасти ваш батальон. На основных уровнях это скорее выжидательная игра, чем бешеная гонка, заявленная в названии, и время иногда тянется неестественно долго. Не сказать, что все уровни одинаковы. Вы то привязаны к месту, держась оборонительной тактики и защищая свою базу от вражеской атаки, а то вас преследует многочисленная

► Много внимания уделено следящим камерам и автоматическому выбору действия, но ни то, ни другое толком не работает.



► Unigine взаправду любит большие взрывы и густые клубы дыма. Не знаем, с чего мы так подумали. Считайте, что это просто нух.

армия, и решения приходится принимать на ходу; или надо маневрировать между снующих кораблей.

Oil Rush – детище восточно-европейского разработчика Unigine Corp, производителя одноименного графического движка в тестовой версии *Heaven* под Windows, так что яркие визуальные впечатления от игры вполне ожидаемы. Во-первых, это, конечно, ее прозрачное море; но и все объекты, находящиеся на нем, поражают своей детальностью, хотя, надо признать, вариантов их немного. Одни и те же платформы и боевые единицы кочуют с уровня на уровень, и маринистические пейзажи в итоге приедаются.

Маленькое чудо

Честно говоря, все это графическое великолепие, может быть, и не так важно. Вы же знаете, что мы любим? Глядеть в маленькую карту, скромно уместившуюся в нижнем правом уголке монитора, примерно на одну его восьмую. *Oil Rush* потакает нашим слабостям. Такое впечатление, что в этой игре весь интерфейс вывернут наизнанку: мини-карта, занимающая ничтожно малую часть экранного пространства – самый важный и полезный компонент этой игры.

Не будет преувеличением сказать, что легко пройти любой уровень, глядя только на карту. Взгляда мельком на большой экран, лишь чтобы ее обновить, вполне достаточно. В описании игры всячески подчеркивалось значение всяких камер слежения, но мы ими почти не воспользовались. Еще один повод придаться: *Oil Rush* должен был выйти еще в 2010 г. Бета-версия появилась в марте 2011. Так откуда же

в официальном релизе ошибки правописания? Грамматические странности в озвучке еще объяснимы, ведь все обычно записывается в самом начале и намертво, да и Unigine – не из англоговорящей страны. Но ошибки в описании миссий следовало отловить и устранить первому же из команды бета-тестеров.

Впрочем, простим им это. Закусив губу, если нужно. Ведь *Oil Rush* – игра симпатичная, один из лучших образцов обоих жанров в Linux – и стратегия, и обороны крепости. Несмотря на неудобное решение с мини-картой. Жанровое своеобразие игры достойно многих похвал: мы бы сказали, что для новичка она гораздо доступнее, чем обычная стратегия, и гораздо увлекательнее, чем нескончаемое отражение атаки за атакой на крепость. И поскольку она доступна на разных платформах, при желании вы легко найдете вариант с мультиплеером. А желание будет. **LXF**

LINUX FORMAT Вердикт

Oil Rush 1.07

Разработчик: Unigine Corp
 Сайт: www.oilrush-game.com
 Цена: \$19,99

Сюжет	8/10
Графика	7/10
Длительность	8/10
Оправданность цены	7/10

» Обманчиво глубокая и непредвиденно причудливая, *Oil Rush* кишит ценными идеями.

Рейтинг 8/10

XBMC 11 Eden

Бен Эдвард попадает в Эдем, точнее в *XBMC 11 Eden*, и подтверждает, что там действительно хорошо.

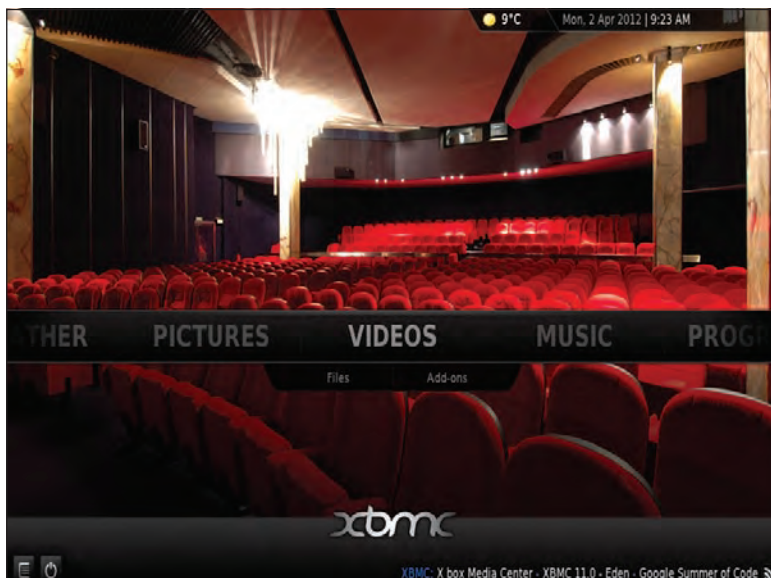
Вкратце

» *XBMC* — это медиа-плеер и центр развлечений. См. также: *VLC*, *Mythbox* и *Totem*.

Последняя версия *XBMC* выходит с поддержкой Windows, Mac OS X, iOS, Apple TV и Linux — что называется, на любой вкус. В документации релиза сообщается о 106 новых или обновленных функциях, от доработки пользовательского интерфейса до улучшения поддержки различных архитектур процессора. Так что смысл обновиться есть для всех.

Мы протестировали *XBMCbuntu* с live DVD, специально собранный командой, чтобы выгодно представить новшества. Диск сделан на основе *Lubuntu*, и загружается сразу в приложение *XBMC* — прямо из коробки. Ну, или, по крайней мере, так было задумано — мы обнаружили, что звук был слишком тихий, чтобы наслаждаться просмотром видео в полной мере.

В аудиоменю категории видео нашлась функция Усиление звука, но хоть он и стал громче, качество его заметно снизилось. Окно Audio Вывод в Настройках также не помогло. Решение нашлось через регулятор Настройки Звук в окне Програм-



» *XBMC* сменил традиционный рабочий стол на новый интерфейс, более приличествующий современным медиа-устройствам.

«Чуть не забыли сказать: он запускается на Raspberry Pi.»

мы > *XBMC* > Аудиомикшер. Зачем было управление звуком разносить в три разные функции, непонятно.

На главном экране семь опций: Погода, Изображения, Видео, Музыка, Программы, Воспроизвести диск и Система. Нам пред-

ставляется, что популярнее всего будут Музыка и Видео. И там, и там есть функция просмотра соответствующих файлов на данном компьютере, а также возможность установки модулей расширения для просмотра содержимого разных интернет-ресурсов. В общем, все работает неплохо — правда, некоторые приложения оказались между двух стульев: например, Facebook Media, позволяющий просматривать видео и изображения из этой популярной сети в *XBMC*, не попал ни в одну из категорий и оказался внизу меню Программы.

Несколько расширений категории Видео в нашей системе не функционировали — подозреваем, из-за того, что хосты фильтруют IP-адреса из некоторых стран. К счастью, есть сторонние дополнения, позволяющие смотреть британское ТВ, такие как BBC Player и 4OD. Установить их проще простого: скачиваете ZIP-файл, а затем идете в Система > Дополнения > Установить из Zip-файла и выбираете свежескачанный архив. Фильтр по местоположению пользователя — это не вина *XBMC*, такие ограничения налагают хосты самих потоков. Если вы не нашли искомое в официальных репозиториях, загляните на http://wiki.xbmc.org/index.php?title=3rd_party_add-on_repositories, там выбор больше.

О достоинствах этой программы можно судить и по тому, что наше главное наре-

кание касалось раскладки. Интерфейс великолепен, а экран прекрасно настроен для удаленного управления — никаких лишних кнопок, можете вводить текст любой «указкой» — только неясно, почему для экранной клавиатуры разработчики оставили выбор на алфавитной раскладке, а не на QWERTY.

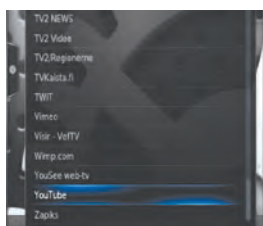
Ах, да, чуть не забыли сказать: приложение запускается на Raspberry Pi. Так что если у вас в руках есть это маленькое устройство, можете превратить любой «ящик» в домашний кинотеатр за какие-то пустячные \$35. **LXF**

Свойства навскидку



Быть в курсе

Теперь можно получать прогноз погоды по местоположению (на основе IP-данных).



Делайте выбор

Расширения позволяют получить доступ ко многим сетевым видеохранилищам.

LINUX FORMAT Вердикт

XBMC 11 Eden

Разработчик: Team XBMC
Сайт: www.xbmc.org
Цена: Бесплатно

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

» Идеален для просмотра потокового видео и создания домашнего кинотеатра... становится, стоит лишь обратиться в меню.

Рейтинг **8/10**

Gnome 3.4

Год спустя после выхода Gnome 3.0 **Джонатан Робертс** вновь видит, что рабочий стол пытается реализовать свой потенциал.

Вкратце

» Рабочий стол Gnome дружелюбен к касаниям, но также предоставляет много быстрых кнопок для обычных пользователей. См. также: KDE, XFCE и Unity.

Вышел Gnome 3.4, третий стабильный релиз в серии 3.x. И нам не терпелось его попробовать, ведь прошлый год выявил в Gnome 3 значительный потенциал и всего пару острых углов. И первое, что нужно отметить – Gnome 3.4 появился не только чтобы их сгладить, но и чтобы представить несколько долгожданных новых функций. Больше всего сил было вложено в web-браузер по умолчанию, *Web* (также известный как *Epiphany*).

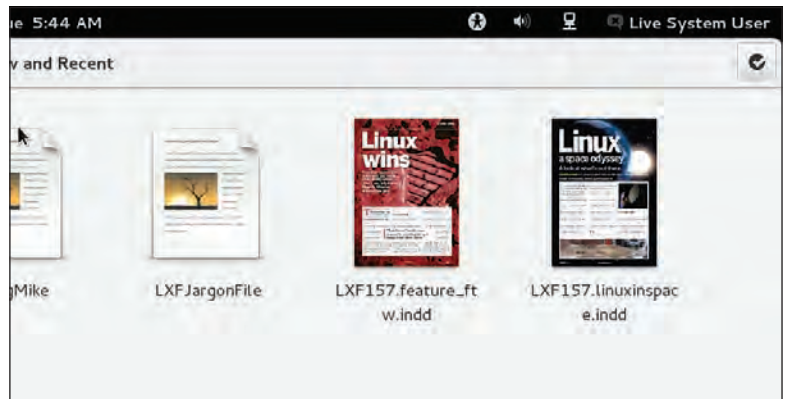
Благодаря движку *Webkit*, он быстро работает и влет проходит тест *Acid3*. Кроме того, новый движок адресной строки *SQLite* также значительно увеличил быстродействие этого элемента интерфейса. Очень жаль, что многие пользователи даже не смогут оценить эти улучшения, поскольку в большинстве дистрибутивов *Web* по умолчанию заменяет на *Firefox*. А между тем, его внешний вид тоже претерпел коренные изменения, и стал великолепен. Здесь теперь нет ничего лишнего: никаких строк состояния, никаких строк меню. Все минималистично, в духе *Chrome* – то, что нужно для браузера.

Никаких заголовков

Web запускается сразу в полный экран, как и в *Unity*, без строки заголовков, оставляя вам максимум свободного места. На первый взгляд немного непривычно, но мы, на нашем тестовом ноутбуке, освоились довольно быстро.

Многие изменения интерфейса коснулись и других приложений, впервые представленных в Gnome 3.2, включая Документы и Контакты. Из не самых заметных – появление поиска по документам в меню Обзор. Ранее это был поиск

» Обзор теперь использует *Tracker*, чтобы осуществлять поиск по файлам.



» Интегрированный с *KVM*, *Voxes* обеспечивает удобство виртуализации.

только по недавно использованным, а теперь, при помощи *Tracker*, осуществляется полный. Это кажется настолько естественным, что просто не верится, как этого могло не быть в первых релизах Gnome 3.

Больше нового ПО

Помимо тех, что были представлены в Gnome 3.2, в 3.4 появилось еще одно новое приложение. Оно называется *Voxes* и создано для обеспечения удобного интерфейса удаленного доступа к другим машинам. То есть, оно сочетает возможности виртуальной машины (используемой с *KVM*) с программой удаленного просмотра.

Virtualbox – это прекрасно, но постоянно копаться в модулях ядра довольно утомительно; с другой стороны, до появления *Voxes* запускать *KVM* приходилось из командной строки или при помощи менеджера виртуализации *virt-manager*.

Шибко продвинутых функций здесь ожидать не приходится, но зато вы сможете быстро и легко пробовать новые релизы ваших любимых дистрибутивов. С таким обилием новых и улучшенных программ, некоторые старые приложения Gnome выглядят обветшалыми. Пора бы *Rhythmbox*, *Brasero* и *Evolution* тоже примерить новый интерфейс; пока этого нет, в Gnome 3 все равно будет ощущение незавершенности.

Пользователи Gnome со стажем наверняка задаются вопросом, были ли, наконец, устранены некоторые раздражающие «свойства» Gnome 3. Отвечаем: не были. До сих пор нужно нажимать Alt,

чтобы выключить компьютер (наш выпускающий редактор, новичок в Linux, счел это смехотворным). Alt+Tab все так же переключает приложения, а не окна, а кнопка запуска в *Dash* по-прежнему возвращает вас в уже открытое приложение вместо запуска нового экземпляра.

После трех стабильных релизов, похоже, пора уже перестать надеяться, что разработчики вернуться к старому. Но эти мелкие неудобства уже не так заметны, благодаря увеличению количества и качества доступных расширений Gnome. Последние помогут перенастроить многие элементы, и легко устанавливаются через web-браузер. Таким образом, этот релиз продолжает укреплять потенциал Gnome 3. Хорошо бы только поумерить пыл с новшествами и обратить внимание на устаревшие элементы. **LXF**

LINUX FORMAT Вердикт

Gnome 3.4

Разработчик: Gnome Foundation

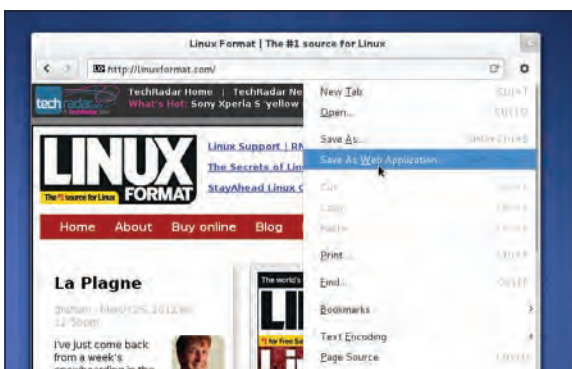
Сайт: www.gnome.org

Лицензия: Бесплатно под GPL

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

» Прогресс за последний год налицо, но часть приложений по умолчанию лучше бы обновить.

Рейтинг 7/10



Audacity 2

Грэм Моррисон из всех сил пытается компенсировать проклятия, прозвучавшие по его поводу на TuxRadar.

Вкратце

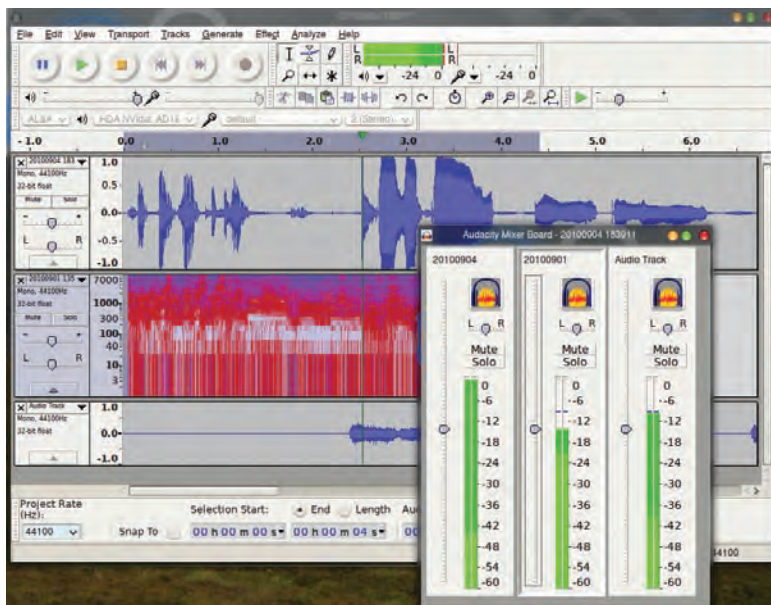
» Редактируйте аудиофайлы, применяйте эффекты и экспортируйте в разные форматы. См. также: *Jakosher* и *Ardour*.

Audacity – неувядающий свободный аудиоредактор. Это прекрасный инструмент для редактирования музыки, вставки звуковых файлов, сведения подкастов и улучшения записей. И все это он умеет уже давно. Как и *GIMP*, Audacity годы просидел «в ожидании стабильного релиза». И, в отличие от первого, таки дождался. Версия 2.0 появилась в середине марта, почти 10 лет спустя выхода первой.

Результатом столь долгого периода созревания стало несколько новых функций, по сравнению с прошлыми итеративными релизами до 1.3. Переписаны многие эффекты, обновлена панель инструментов. Однако графический интерфейс остался без изменений, равно как и общий подход; нового движка для функций также не появилось. Вместо этого, версия 2.0 – развитие старой, хотя и превосходно реализованной идеи.

Приступая к правке

Большую часть времени в Audacity вы будете проводить, редактируя музыкальные файлы, каковой процесс постепенно оттачивался, чтобы к этому релизу стать, пожалуй, самым интуитивно понятным из всех альтернативных приложений, с которыми нам приходилось иметь дело, коммерческих или свободных. Выделение области, ее редактирование, вырезание, вставка и обработка практически безупречны. Все очень быстро и просто. Выпадающее меню, переключающее из привычного вида звуковой амплитуды на спектрограмму или даже микшерский пульт с поддержкой



» Немногие аудиоредакторы, по любой цене, умеют отображать спектральные данные и редактировать несколько дорожек параллельно.

измерения уровня звука – главный секрет Audacity. Но еще более впечатляет возможность параллельного редактирования нескольких дорожек, с последующим сведением. Например, если вы делаете 45-минутную запись какого-нибудь несдержанного технаря со склонностью к ругани, можно вырезать всю нецензурщину из каждого трека простым выделением мышью. С новой функцией Sync-lock для поддержки пометки треков это стало еще проще – такого мы не видели ни в одном редакторе, что дает Audacity сто очков вперед. Не хватает разве что масштабирования звуковой волны при помощи полосы прокрутки, но здесь ту же функцию выполняют клавиши быстрого доступа.

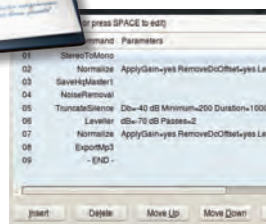
К сожалению, нельзя сказать того же о применении эффектов. Звучат они нормально, но жутко неудобный способ их наложения, даже в функции предпросмотра, безнадежно устарел. Сначала нужно выделить область, потом открыть эффект, выбрать его, нажать предпрослушивание и надеяться на лучшее. Как-то это не интуитивно и не музыкально.

Было бы здорово увидеть здесь контрольную панель, с которой можно было бы перетаскивать те или иные эффекты на аудиопоток, прежде чем применить их ко всему файлу. Импорт и экспорт файлов улучшены, а добав-

ленная поддержка AC3/M4A/WMA пригодится для записи на мобильный телефон. Также полезным мы сочли редактор метаданных и возможность сохранять подробности. Качество модулей экспортирования MP3 и Ogg можно слышать на наших подкастах, хотя нам всегда не везло со встроенным подавлением шума – оно слишком примитивно для должной работы. В целом же, Audacity – прекрасный инструмент редактирования, и мы надеемся, что с выходом версии 2.0 подход к работе с эффектами все-таки поднимется над прошлым веком. LXF

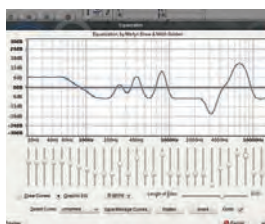


Свойства навскидку



Обработка пакетом

Конвертируйте файлы и добавляйте эффекты при помощи одной подробной диалоговой формы.



Классные эффекты

Некоторые эффекты великолепны – например, эквалайзер; хорошо бы их было проще применять.

LINUX FORMAT Вердикт

Audacity 2.0

Разработчик: The Audacity Team
Сайт: <http://audacity.sourceforge.net>
Лицензия: Свободно под GPL v2

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

» Лучший аудиоредактор на сегодняшний день, но, пожалуйста, пусть разработчики улучшат эффекты!

Рейтинг 8/10



Renoise 2.8

Этот номер журнала опять едва не опоздал к выпуску – на сей раз потому, что Грэм Моррисон увлекся, издавая электронные звуки и восхищенные “Wow”.

Вкратце

» Потрясающий саундтрекер, идеально подходит для повторяющейся игровой музыки. См. также: свободный *Shism Tracker*.

Э то третий по счету основной релиз *Renoise* со времен нашего обзора версии 2.0 в LXF117. Если вы успели приобрести *Renoise* еще тогда, ваша лицензия будет продолжать действовать, дела покупку очень выгодной. А для тех, кто впервые слышит это название, сообщаем: *Renoise* – трекерный редактор, то есть инструмент для создания музыки, использующий числовую матрицу для представления каналов, нот, звуков и эффектов. Некогда (в середине 1980-х) это был самый эффективный способ втиснуть музыкальные данные в жесткие рамки компьютерной среды, и тот характерный звук мы прекрасно помним по играм и демо-сценам.

И сегодня, в 2012-м, трекеры не сдают позиций, ведь с их помощью сочинители могут контролировать любой аспект своей музыки. Это все равно что *Vim* или *Emacs* для программистов. Но сказать, что *Renoise* – лишь скромный саундтре-



» Среди прочих новых функций – быстрая загрузка больших сэмплов, улучшение структуры паттернов и новые возможности визуализации.

«ЛУЧШИЙ НОВЫЙ ЭФФЕКТ НЕ СВЯЗАН СО ЗВУКОМ. ЭТО МЕТА-МИКСЕР.»

кер, значит не сказать ничего. Да, это все тот же матричный принцип, но обогащенный всеми аудионновациями XXI века, включая цифровые эффекты, автоматизацию, звуковые ресурсы, библиотеки сэмплов, контрольную маршрутизацию и визуальное представление сигнала. Все это превращает *Renoise* в полноценную музыкальную студию.

Pulse не слышен

Чтобы установить *Renoise*, просто скачайте готовый двоичный архив и запустите установочный скрипт, и все тут же заработает. *Renoise* сам подтянет свои звуковые библиотеки и пресеты и подключится к звуковой системе ALSA. Правда, чтобы хоть что-нибудь услышать, нам пришлось убить *PulseAudio*, но мы это сделали с радостью, и теперь по крайней мере есть возможность реинициализировать аудио из панели Настроек.

Немного поотвыкнув от *Renoise*, очень приятно вновь увидеть его потрясающе функциональный интерфейс. Введя несколько нот на пару каналов и нажав

на Воспроизведение, чувствуешь себя хакером в середине 1980-х, потому что все панели начинают двигаться, прыгать и подстраиваться под музыку. Интерфейс безупречен.

Нам больше всего пришлось по душе новая возможность создавать имена-псевдонимы в редакторе паттернов. Именно там вы соединяете блоки данных в цельную мелодию, а псевдонимы позволяют их дублировать. Но здесь, в отличие от обычного копипаста, любые изменения, применяемые к одному блоку, автоматически применяются к другим, подобно символической связи в файловой системе Linux. Это значит, что вы можете в черновом виде записать секцию ударных, разместить ее в нужные блоки, а затем вернуться к ней вновь, чтобы отредактировать.

Автоматизация эффектов и звуков не задета, и вы по-прежнему можете вносить изменения в звук, рисуягибающие трека. Вы также можете сводить несколько треков в один паттерн для удобства редактирования, и сворачивать их, подобно сворачиванию функции в редакторе исходного кода.

Великолепные эффекты теперь дополнены функцией Multitar delay [комбинация задержки, эха, фильтра и реверберации, – прим. пер.] с синхронизацией темпа. Появился и новый повторитель, способный добавлять мастер-эффекты, а также возбуждатель, идеально подходящий для

основного канала, чтобы добавить лоску 8-битным сэмплам. Однако лучший новый эффект вовсе не связан со звуком. Это мета-микшер. Он управляет мета-контроллерами, объединяя до трех контрольных выводов разных модулей, прежде чем использовать его собственный выход для контроля еще одного, например, инструментального, или любого другого. Таким образом *Renoise* превращается в модульный синтезатор-сиквенсер с безграничными возможностями.

Renoise – феноменальный инструмент композитора, и если вам хоть чуточку интересно попробовать приписать ноту к ноте, он стоит затрат. LXF

LINUX FORMAT Вердикт

Renoise 2.8

Разработчик: E Mueller and Z Tesic
Сайт: www.renoise.com
Цена: €58

Функциональность	11/10
Производительность	11/10
Удобство использования	9/10
Документация	11/10

» Вот что получается, если в душу саундтрекера вложить студию звукозаписи.

Рейтинг 11/10

Ядро Linux 3.3

Игорь Штомпель благоговейно вступает в святилище операционной системы. Какие вести придут к нам оттуда?

Вкратце

» Сердцевина ОС Linux, по сути определяющая функциональные возможности системы. См. также: BSD, Mac OS X.

Ядро – это основная часть операционной системы, которая по сути определяет ее возможности. Его новые выпуски для ОС Linux всегда приносят большое количество изменений. Вот, например, как версия 3.3...

Выпуск ядра версии 3.3 стал знаменательным событием – его исходный код превысил 15 млн строк. Но одним количеством строк исходного кода достоинства этого ядра не ограничиваются. Давайте рассмотрим подробнее то, что в нем еще нового.

Основные изменения

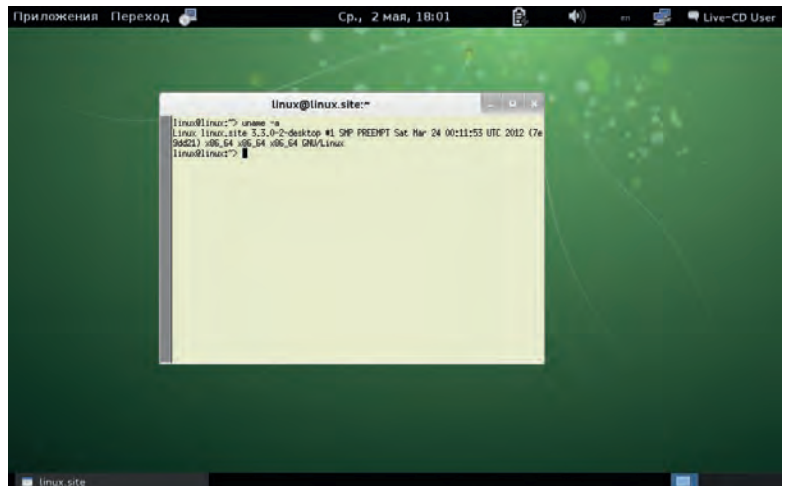
Переход дистрибутивов на новое ядро только начался. Так, проект openSUSE на ядре ветки 3.3 строит свою новую версию дистрибутива. На момент написания обзора была доступна версия openSUSE для разработчиков – 12.2 build 0318 (Milestone 3). Текущая версия Arch основана на ядре 3.3.4. Следующая за Fedora 16 разрабатываемая версия проекта основана на ядре 3.4.0. На этой же версии ядра основан и будущий релиз Ubuntu – 12.10.

Разработчики ядра 3.3 осуществили интеграцию наработок, реализованных в рамках проекта Android. Как известно, создатели последнего подготовили ряд улучшений для ядра Linux, которые они

«Количеством кода достоинства этого ядра не ограничиваются.»

использовали в своем проекте. Данные наработки долгое время не были интегрированы в ядро Linux из-за разногласий разработчиков ядра Linux и проекта Android. В частности, речь идет о таких технологиях, как `ashmem` (разделяемая память, которая дает возможность маркировать незакрепленные страницы памяти – если возникает ситуация нехватки свободной памяти, данные страницы можно освободить), драйвер для ведения системного журнала и других.

Еще одним важным новшеством стала реализация поддержки архитектуры C6X процессоров компании Texas Instruments. На базе данной архитектуры выпускаются одноядерные и многоядерные процессоры семейства DSP. Кстати, ранее



» openSUSE 12.2 (64 бита) для разработчиков на базе ядра версии 3.3.

поддержку этой архитектуры обеспечил проект `linux-c6x.org`.

Улучшена реализация операций балансировки ("balance") и поддержки RAID в файловой системе Btrfs. Балансировка в Btrfs представляет собой перезапись данных и метаданных файловой системы. Для чего это необходимо? Вот, например, добавляется новый жесткий диск, а операция балансировки позволит перераспределить данные файловой системы на него. С другой стороны, такая операция может занять много времени. Новшеством стало то, что процесс балансировки теперь возможно приостанавливать и возобновлять. А сам процесс балансировки стал основываться на базе профилей, что позволяет выполнить балансировку только для пустых групп блоков.

Что касается улучшения поддержки RAID, то реализована возможность перехода с одного RAID-уровня на другой, путем перестроения [restripe] данных. Для выполнения указанных операций можно задействовать инструментальный `btrfs-progs` из ветки `parser`.

Значительные улучшения произошли в сетевой подсистеме ядра. Разработчики ядра интегрировали Open vSwitch – программную реализацию многоуровневого (с поддержкой большого количества протоколов) сетевого коммутатора. Надо отметить, что в ядре уже имеется реализация виртуальных сетевых мостов, но Open vSwitch ориентирован на решение более комплексных задач: например, в качестве

виртуального коммутатора в окружении сервера виртуализации. Среди поддерживаемых Open vSwitch интерфейсов управления – sFlow, Netflow, RSPAN, CLI. Кроме того, Open vSwitch позволяет подключать программные расширения, а также использовать протоколы Openflow и OVSDB для управления.

Также реализовано новое сетевое устройство teaming, которое призвано стать быстрой, масштабируемой, прозрачной и управляемой из пространства пользователя заменой технологии bonding. Новая технология teaming позволяет создавать виртуальные интерфейсы, объединяющие различные Ethernet-устройства. Это, в частности, позволит повысить пропускную способность, обеспечить отказоустойчивость.

В рамках борьбы с эффектом Bufferbloat – влиянием буферизации ряда на задержки и пропускную способность – установлено ограничение на размер пакета данных, который можно поставить в очередь передачи данных для сетевого устройства. Это позволяет обеспечить приемлемую транзитную задержку для пакетов с высоким приоритетом, причем не понадобится очистка соответствующих аппаратных очередей при появлении данных для отправки.

Для системных администраторов реализованы управления сетевыми приоритетами. Теперь администратор может управлять приоритетом сетевого трафика, который генерируется приложениями

(*cgroup*). С помощью *cgroup* можно назначить процесс группе, которая определяет приоритет для трафика на необходимом сетевом интерфейсе.

Появилась возможность установки ограничения на размер TCP-буфера, что обеспечивает контроль нехватки памяти для протокола TCP. Речь идет о памяти на уровне ядра, а конкретно – той, что предназначается для поддержки соединений на базе протокола TCP.

Важные изменения произошли в реализации поддержки файловой системы ext4. Механизм изменения ее размера теперь полностью реализован в ядре Linux, что значительно ускорило операцию изменения размера. Было проведено тестирование (<https://lwn.net/Articles/466299/>), в ходе которого размер файловой системы ext4 был изменен с 20 ГБ до 230 ГБ за 3,3 секунды, тогда как с помощью утилиты *resize2fs* на это понадобилось более 5 минут.

Разработчики также реализовали поддержку загрузки с использованием EFI (Extensible Firmware Interface) – специального интерфейса между ОС и микропрограммами [firmware], которые призваны управлять оборудованием с помощью функций низкого уровня. Другими словами, EFI поможет произвести инициализацию оборудования, а затем передаст управление загрузчику ОС.

Прочие улучшения

Изменения в ядре весьма многочисленны, и вместить их все в рамки краткого обзора невозможно. Ведь они охватывают поддержку архитектур, драйверов, системных сервисов, управления памятью, файловых систем, сети, виртуализации, криптографии, трассировки и профилирования... Более подробно с новшествами можно ознакомиться в официальном анонсе – <https://lkml.org/lkml/2012/3/18/132>, или на странице портала kernelnewbies.org: http://kernelnewbies.org/Linux_3.3. Мы здесь лишь затронули «верхушку айсберга».

Поговорим о более мелких новшествах, которые показались нам особенно интересными.

Для файловой системы Btrfs стала доступна новая утилита, получившая название *integrity check* – она ориентирована на разработчиков и позволяет осуществлять расширенное тестирование целостности данных. Например, при записи данных на диск утилита следит за тем, чтобы не была записана лишняя информация (поддельные ссылки и т.д.), способная привести файловую систему в нестабильное состояние и обусловить потерю данных.

Несколько слов о поддержке аппаратной части. Реализована поддержка LPAE (Large Physical Address Extension) для процессоров ARMv7, что дает возможность адресовать память более 4 ГБ. Появилась поддержка устройств OpenBlockS 600 на базе процессоров Power PC. Для архитектуры S390 добавлена поддержка физической памяти более 4 ТБ. Для платформы x86 реализована поддержка NumaChip.

Разработчики драйверов также предложили интересные решения, ставшие составной частью ядра Linux версии 3.3. Так, из ветки ядра staging (отдельная ветка ядра для предварительного тестирования кода) в основной код ядра перешел драйвер для интегрированных видеокарт от Intel – GMA500. Реализована начальная поддержка видеокарт NVD9 от NVIDIA. Для устройств Exynos (системы на кристалле, SoC, на базе процессоров ARM от компании Samsung) добавлена поддержка HDMI. Также включены новые USB-драйверы: Samsung Exynos OHCI, Marvell Host Controller, Marvell usb OTG, iPhone 4S (драйвер ipheth). Обеспечена поддержка звуковой карты Cirrus Logic 4213, а также оборудования, которое поддерживает ускорение обработки сжатого звука.

Среди других устройств, драйвера для которых были добавлены в ядро 3.3 – Realtek ALC5632, Lowland, 1277-EV1 Littlemill.

Но указанным поддержка устройств не ограничивается – введена также поддержка различного уровня для устройств: хранения данных, сетевых, ввода, web-камер и др. Как видно, перечень аппаратуры серьезно расширен.

Заключение

Выход ядра Linux версии 3.3 принес изменения, которые ожидались давно. В частности, осуществлена интеграция с работами проекта Android. Значительно улучшена реализация сетевой подсистемы (например, интеграция в ядро Open vSwitch). Реализация Btrfs стала более гибкой (достаточно вспомнить возможность перехода с одного RAID-уровня на другой). Наиболее заметным в области поддержки аппаратной части стала реализация разработчиками поддержки архитектуры C6X процессоров от компании Texas Instruments. Что касается файловой системы ext4, то она получила возможность быстрого изменения размера, далеко опередив ранее доступные механизмы для этой цели. Расширить возможности процесса загрузки теперь можно с помощью технологии EFI.

Большое количество мелких изменений и усовершенствований сделали ядро Linux лучшим и более гибким. **LF**

LINUX
Вердикт

FORMAT

Linux Kernel 3.3

Разработчик: Проект Linux kernel
Сайт: <http://www.kernel.org>
Цена: Бесплатно под GPL

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

» Эволюционная версия, принесшая много новшеств, в том числе в поддержке устройств. Интеграция с нарбаботкой Android – яркий пример.

Рейтинг 9/10

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 обслуживание 250 ПК



65 240 руб.

DR.WEB OFFICE SHIELD NEO

Рассчитан на обслуживание 50 ПК



Linux center

www.linuxcenter.ru



WWW.LINUXCENTER.RU/SHOP/ANTIVIR/

PocketBook A 10"

Артем Зорин протестировал новый планшетный компьютер от PocketBook.

Вкратце

» Маломощный планшетный компьютер, сочетающий функции электронной книги и Android-устройства без претензий на «мейнстрим».

Сейчас стало модно выпускать планшетные ПК. Каждая мало-мальски уважающая себя компания норовит выпустить свой планшет. Но успех ждет далеко не многих. Сегодняшний экспонат – PocketBook A 10", созданный в недрах международной компании PocketBook International S. A. и имеющий украинские корни (разработка программного обеспечения осуществляется в Киеве) – отличный пример того, как можно сделать хороший планшетный ПК за умеренную цену.

Внешний вид

Начнем с дизайна: он, как и у многих подобных устройств, предельно прост и лаконичен, современен. Единственным минусом внешнего вида PocketBook A 10" можно считать совершенно безвкусную белую окантовку вокруг корпуса. Уж лучше бы сделали его либо полностью черным, либо белым. Белая окантовка – это зло! Производитель, убери ее, пожалуйста. Она скользит в руках и выглядит ужасно. Или хотя бы прорезинь этот белый ужас...

Остальные детали дизайна не вызывают отторжения. Отметим, что пластик и стекло, использованные в корпусе устройства, приятны на ощупь и не скользят в сухих руках. Вес планшета – один из главных факторов в повседневной работе. И тут PocketBook A 10" слегка подкачал: он слишком тяжел для такого относительно слабого устройства, его вес 670 грамм.

В PocketBook A 10" реализован легкий в использовании орган управления для максимального удобства при чтении. Он отличается от стандартных планшетов на базе Android тем, что кроме привычных клавиш «Домой», «Меню», «Назад» и «Поиск», на ней расположены две кла-



» Кнопка включения питания имеет склонность «западать».



» Планшетный компьютер PocketBook A 10 сочетает приемлемую цену и такое же качество. Могло быть и лучше, но за другие деньги...

виши перелистывания страниц. Все клавиши оснащены приятной мягкой подсветкой. На левой боковой панели помещается кнопка регулировки уровня звука, который не особо удивляет своей мощностью, но довольно чист. В общем, звук как звук.

Там же, на левой боковой панели, расположены кнопки регулирования уровня звука. Не самые удобные – при нажатии они слегка продавливаются внутрь корпуса, что при частом использовании вызывает известное раздражение.

Рядом с кнопками регулировки звука расположен стандартный разъем типа Jack 3,5 мм для подключения наушников или гарнитуры. На правой боковой панели разместились не слишком удобная кнопка включения/выключения питания (она тоже имеет склонность продавливаться вовнутрь при частом использовании), а также слот для карт MicroSD емкостью

до 32 Гб и Micro USB (скорость передачи данных до 12 Мб/с).

На лицевой панели имеется слабая камера, годная лишь для разговоров по Skype (матрица 2 млн пикселей), и стереодинамики (2×0,5 Вт), довольно громкие. Но для такого дешевого устройства (цена стартует от 10990 рублей) это нормально. А вот кнопки управления устройством порадовали удобством расположения и приятным ходом. Листать электронные страницы реально удобно!

Железо

Пройдемся по железу. Тут все более или менее стандартно для подобных устройств класса low-end: цветной мультисенсорный дисплей с диагональю 10 дюймов (разрешение 1024×768 пикселей) – «это уникальное лицо модели PocketBook A 10», гордо гласит реклама на сайте производи-

теля. Нам это лицо показалось каким-то не совсем уникальным, учитывая, что технологии, использованные в его производстве, уже 20 лет как не новинка... Что порадовало в дисплее, так это его соотношение 4:3 – это здорово, когда вы много читаете. В итоге: экран подопытного идеален для чтения и просмотра фотографий, сайтов, но не совсем подходит для широкоформатного видео.

Есть стандартный для планшетов акселерометр и модуль Wi-Fi, работающий как часы – четко и без сбоев. Во время теста планшет без проблем подключался ко всем сетям, имеющимся в округе.

Емкость встроенной литиево-полимерной аккумуляторной батареи составляет 6600 мАч. Если эту цифру перевести во время пользования ридером, мы получим 4 часа в смешанном режиме полной нагрузки (включенный Wi-Fi и Bluetooth, web-серфинг, чтение, прослушивание аудио, просмотр видео) или 2 часа в режиме непрерывного просмотра видео.

Модель PocketBook A 10" отличает привычная для всех устройств PocketBook мультимедийность. Устройство поддерживает 11 текстовых форматов: PDF (Adobe DRM), EPUB (Adobe DRM), PDF, EPUB, FB2, TXT, DJVU, RTF, HTML, DOC, DOCX, четыре графических – JPEG, BMP, PNG, TIFF, три формата видео – AVI, MKV, MP4 и два формата аудио – MP3, WAV. Это неоспоримый плюс устройства.

«В целом, устройство получилось быстрым и симпатичным.»

Для обмена информацией в PocketBook A 10" есть встроенный Bluetooth 2.1. Интернет-соединение обеспечивается через Wi-Fi 802.11 (b/g/n). Огорчило отсутствие модуля GSM, ибо Wi-Fi есть далеко не везде, а в Интернете хочется быть всегда, даже во сне; также хотелось бы иметь модуль GPS (а то и ГЛОНАСС!).

PocketBook A 10" работает на базе ARM-процессора TI OMAP 3 (1-ГГц Cortex A8). Быстродействие устройства дополнительно обеспечивают 512 МБ оперативной памяти; для хранения данных пользователя доступно 2 Гб постоянной памяти из имеющихся 4.

Расстроило наличие всего лишь одной камеры на фронтальной стороне, да и ее расположение не самое удобное – справа внизу. Качество видео и фотографий соответствуют предназначению камеры. Но для передачи ваших фотографий с очередного митинга или пикета/голодовки в Twitter или Instagram камера вполне сойдется, при условии, что будет Wi-Fi...

» Перелистывать электронные страницы удобно, благодаря кнопкам со стрелками.



Программы

Устройство поддерживает функцию преобразования текста в речь Text-to-Speech и прочтет книги на 4 языках из 27 доступных для скачивания на сайте производителя. Русский язык у «говорителя», как всегда, хромает, поэтому соотечественникам пользоваться этой функцией можно посоветовать исключительно смеха ради, а иностранцам вовсе не стоит – просто стыдно за великий и могучий.

Планшет работает на платформе Android 2.3.5. Обновлений прошивки при тестировании обнаружено не было, что тоже слегка расстроило: хотелось бы попробовать Android Honeycomb.

Из предустановленного ПО можно особо отметить браузер *Dolphin*, как две капли воды похожий на *Google Chrome* (Adobe Flash нужно скачивать отдельно из Android Market, ныне Google Play), и офисный пакет *Office Suite Pro 5*. С его помощью можно работать со множеством документов. Из особенностей программы стоит отметить поддержку форматов DOC, DOCX, TXT, XLS, XLSX, CSV, PPT, PPTX, PPS, PPSX и PDF, встроенный файловый менеджер, возможность работы с Google Docs. PocketBook A 10" поддерживает онлайн-синхронизацию ваших данных с учетной записью Google. С помощью облака синхронизируются:

- » почтовый ящик Gmail;
- » фотоальбом Google Picasa;
- » контакты и календарь с учетной записью Google;
- » подписки в Google Reader (при установке приложения *Google Reader* на устройство).

Имеется встроенный клиент электронной почты, сильно смахивающий на Mail в iOS.

Самое интересное во встроенном ПО – это Obreey Store, аналог Amazon Store или Google Play от наших украинских друзей. Obreey Store – это новая многоязычная контент-площадка от PocketBook, реали-

зованная с использованием современных технологий.

В ассортименте Obreey Store – книги различных жанров, обучающие детские программы, учебники и справочная литература для школ и ВУЗов (!), словари ABBYY® Lingvo, Collins, Universal и другие, периодика, аудиокниги, тематические подкасты. Короче, есть из чего выбрать...

Остальные приложения нет смысла описывать: все стандартно для Android-планшетов.

Выводы

Скорость работы планшета не вызывает нареканий: приложения открываются быстро, HD-видео не тормозит, большие графические и текстовые файлы (от 5 МБ и выше) открываются на ура и без каких-либо тормозов. В принципе, 512 МБ ОЗУ сейчас не ахти как много, но этого, на удивление, хватает для комфортной работы.

В целом, устройство получилось быстрым и симпатичным, несмотря на белую окантовку. Можно порекомендовать его всем, у кого есть постоянный доступ к Wi-Fi-соединению и кому не хватает бюджета на более мощные и продвинутые устройства. **LXF**

LINUX
FORMAT
Вердикт

PocketBook A 10"

Разработчик: PocketBook International S.A.
Сайт: www.pocketbook-int.com/ru/
Цена: от 10990 руб.

Функциональность	9/10
Производительность	10/10
Удобство использования	7/10
Оправданность цены	7/10

» Хорошая «читалка» с некоторыми функциями планшетного компьютера.

Рейтинг 9/10



ORACLE ПРОТИВ GOOGLE — ПОКА 1:0

Противостояние усиливается

Присяжные признали, что Google в 2 из 5 пунктов нарушил патенты на Java.

Первая фаза судебного разбирательства, рассматривающего обвинение в нарушениях связанной с Java интеллектуальной собственности и патентов компании Oracle в платформе Android, приобрела неприятный для Google оборот. В противоположность недавнему решению Европейского суда, признавшему, что API и языки программирования не могут быть отнесены к категории интеллектуальной собственности, присяжные заседатели при рассмотрении дела в американском суде встали на сторону Oracle и постановили, что компания Google нарушила интеллектуальную собственность, связанную с Java API.

Нарушение прав Oracle признаны по 2 пунктам из 5. В решении указано, что связанных с документацией нарушений не обнаружено, но присяжные признают, что имеет место копирование кода одной из функций (9 строк кода), что может рассматриваться как нарушение интеллектуальной собственности. Присяжные также согласились с тем, что Android API копирует структуру, последовательность и организацию Java API. Однако заседатели не смогли достигнуть единства мнений и однозначно ответить на главный вопрос: выходит ли использование Java API в Android за рамки условий «добросовестного использования [fair use]». Более того, в разбирательстве не был получен ответ на главный и концептуальный вопрос — относится ли API к объектам авторского права. Также в процессе заседания судья позволил себе предположить, что API может подпадать под действие авторского права, и допустил, что процесс создания API может быть сравнен списанием путешественника по проезду из одного города в другой, что не соответствует действительности в области ПО. Подобная двойственная позиция позволила Google настаивать на пересмотре решения, так как вынесен неполный вердикт и была допущена судебная ошибка. Обсуждение данной претензии состоится в ближайшие дни.

Позиция Oracle сводится к тому, что Google скопировала определение 37 Java API из базовых Java-библиотек, считая, что создание API сродни написанию музы-



» Последний CEO Sun — Джонатан Шварц. Он признает правоту Google.

ки, не является просто голый идеей и требует ресурсов на экспертизу и разработку. Поэтому, по мнению Oracle, API может рассматриваться как интеллектуальная собственность. В ответ Google утверждает, что нарушения интеллектуальной собственности в Android не было, так как при создании платформы была использована собственная независимая реализация API, без копирования кода библиотек Java, а также были соблюдены условия «добросовестного использования» Java API. Google также подчеркивает, что использование Java API носило «преобразующий» характер, а не «производный», так как на базе данного API была создана принципиально иная технология. Кроме того, юристы Google указали на то, что компания Sun Microsystems публично одобрила способ использования Java в Android.

Напомним, что на первом этапе судебного разбирательства рассматривается только вопрос нарушения связанной с Java API интеллектуальной собственности. Если присяжные признают факт нарушения интеллектуальной собственности Oracle, то заседание перейдет на второй этап, на котором будут рассмотрены вопросы, связанные с нарушением двух патентов. Далее, на третьем этапе будут рассмотрены остальные вопросы, включая размер нанесенного ущерба и наличие умысла.

Примечательно, что победа в разбирательстве с Google может отрицательно сказаться на продвижении самой платформы Java, так как в случае признания принадлежности прав на Java API проект Java окажется не таким открытым, как предполагалось. Непонятно, насколько после такого решения разработчики будут вольны с использованием Java API и в каких ситуациях потребуется получение лицензии от Oracle. По мнению профессора права Тэйлора Очоа [Tyler Ochoa], разработчики будут остановлены в реализации нестандартных вещей на базе Java, и возникнет ситуация, при которой Oracle выиграет один бой, но проиграет всю войну. Более того, признание, что API подпадает под защиту копирайта, будет фактически означать возможность монополизации определенных программных интерфейсов и появление рычага для блокирования создания совместимых альтернативных реализаций (под удар могут попасть, скажем, такие проекты, как *Samba* и клиенты для мгновенного обмена сообщениями).

Мнение эксперта: Google прав!

На проходящее в эти месяцы судебное разбирательство, касающееся обвинения компании Google в нарушении связанной с Java интеллектуальной собственности в платформе Android, в качестве свидетеля был вызван Джонатан Шварц [Jonathan Schwartz], бывший руководитель компании Sun Microsystems, который выступил с неожиданным заявлением.

Анализируя положение вещей в момент зарождения платформы Android, юристы Google попросили Шварца высказаться по поводу утверждений Oracle о том, что Java API в то время являлся проприетарной собствен-

ностью Sun. На что он ответил, что не согласен с такой формулировкой: платформа Java уже была переведена в разряд открытых проектов и воспринималась Sun именно как открытый продукт. По словам Шварца, Java API позиционировался как открытый, и компания Sun была заинтересована в привлечении к его развитию как можно большего числа людей. Ограничения касались только имени Java, но так как Google не использовала в названии своей виртуальной машины слово Java, она вольна была использовать элементы Java API, не нарушая при этом интеллектуальной собственности Sun.

ПРОТИВНИКИ ЗАКЛЮЧИЛИ СОЮЗ

Два гиганта объединились

Патентное противостояние Barnes&Noble и Microsoft закончилось созданием совместного предприятия.

Компании Barnes&Noble и Microsoft анонсировали заключение сделки, ставящей точку в связанном с платформой Android патентном противостоянии. Компании заключили стратегическое партнерское соглашение, воплощенное в создании новой совместной компании, название которой пока неизвестно. Новая компания объединит два направления бизнеса Barnes&Noble: электронные книги и университетские книжные магазины.

По данному соглашению, Microsoft вложит в компанию 300 млн долларов, в обмен на долю 17,6 % в акционерном капитале новой компании (Barnes&Noble будет принадлежать 82,4 %). Barnes&Noble в рамках сделки разработает приложение NOOK для Windows 8, которое будет представлять доступ к одному из самых больших в мире каталогов электронных книг, журналов и газет. В итоге Barnes&Noble и Microsoft предоставят студентам и преподавателям доступную и в Windows 8 технологическую платформу для распространения электронных учебных материалов.

Данная сделка ставит точку в патентном споре двух компаний, в котором Microsoft предъявляла претензии к планшету NOOK на базе платформы Android. Новая компания и Barnes&Noble получат требующую выплаты отчислений лицензию на использование патентов Microsoft для NOOK eReader и других моделей планшетов. Напомним, что в настоящее время

большинство производителей заключили соглашения с компанией Microsoft, подразумевающие выплату отчислений за использование патентов Microsoft в Linux и Android. В компании Barnes&Noble отказались заключать соглашение и пытались судебным путем доказать несостоятельность заявлений Microsoft, собрав и представив более 800 доказательств несостоятельности фигурирующих в разбирательстве патентов.

«Данная сделка ставит точку в патентном споре двух компаний.»

Как замечает Флориан Мюллер [Florian Mueller], известный аналитик проблем патентования ПО, партнерское соглашение Barnes&Noble с Microsoft может свидетельствовать о том, что все обвинения Microsoft в злоупотреблениях патентами, сделанные юристами Noble, которые даже не были представлены к рассмотрению в суде, не имели реальной силы и были порождены отчаянием, поскольку у Barnes&Noble нет своего портфеля патентов, который можно было бы использовать для подачи встречного иска. Попытки доказать несостоятельность уже фигурировавших в других делах патентов является долгим и финансово затратным процессом, удачный исход которого



► Штаб-квартира Barnes&Noble в Нью-Йорке. Здесь любят цензуру.

не гарантирован. Также Мюллер отмечает, что важным итогом соглашения является то, что теперь Motorola Mobility остается единственным производителем устройств с Android, вовлеченным в судебную тяжбу с Microsoft. Но скорей всего, если бы Google не приобрела Motorola, соглашение с Microsoft также было бы подписано, как это сделали Samsung, HTC и LG. **LXF**

Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

Широкая сеть представительств в разных городах России

позволит вам оптимизировать процессы логистики и доставки товара



ПОДРОБНЕЕ О ПАРТНЕРСКОЙ ПРОГРАММЕ:
WWW.LINUXCENTER.RU/PARTNER/



» Ваши программные наработки не пропадут даром

ARM и Android: Программирование

Часть 4: Андрей Боровский берется за программирование на ассемблере ARM и пренебрегает правами root.



Наш
эксперт

Андрей Боровский
В 14 лет сломал школьную локальную сеть. И это оказалось только началом.

В этой, завершающей части мы прикоснемся к увлекательному миру программирования на ассемблере ARM, а также рассмотрим программу Android, которая позволит запускать программы Linux на устройствах Android, не имеющих режима root.

Начиная эту серию статей, я писал, что одной из причин для разработки программ Linux под Android может стать стремление познакомиться с особенностями процессоров ARM в сравнительно комфортных условиях. Именно этим мы теперь и займемся.

NEON!= OFF

Концепция, положенная в основу процессоров ARM, изначально выглядела очень красиво. В то время как процессоры Intel обрастали различными костылями, призванными обеспечить совместимость с предыдущими моделями и конкурентами (AMD), процессоры ARM сохраняли компактную и элегантную структуру. И хотя Intel превосходил ARM в производительности, ARM с его низкими затратами энергии и пассивным охлаждением нашел свою нишу в мире встраиваемых устройств. Второе рождение ARM пережил в нынешнюю эпоху смартфонов и других интеллектуальных мобильных устройств, которые требуют сочетания вычислительной мощи ПК и низкого энергопотребления.

Но за все приходится платить. Современные мобильные устройства насыщены графикой, видео и звуком, а для обработки всего этого в режиме реального времени требуются специальные наборы команд. И процессоры ARM, расставшись с изначальной простотой, обзавелись такими наборами. Причем, как это часто бывает в подобных случаях, введенные сначала наборы дополнительных команд оказались недостаточно удобными и функциональными, а отказаться от них было уже нельзя, так как они использовались во многих приложениях. В результате архитектура ARM, как и Intel, обладает теперь несколькими наборами команд для работы с числами с плавающей точкой и массивами данных, которые частично дублируют функции друг друга.

Если вы откроете файл `/system/proc/cpuinfo` (на нерутованном устройстве содержимое этого файла можно получить, например,

с помощью программы *Droid System Suite*), то увидите перечень расширений (fastmult, vfp и другие), поддерживаемых вашим процессором. Эти расширения похожи на расширения Intel (mmx, sse и т.д.), но, разумеется, не тождественны им. Рассматривать систему команд процессора ARM подробно мы не станем. Помимо самой документации, доступной на сайте arm.com, хорошим пособием является книга *ARM System Developer's Guide*, Andrew Sloss et al. Morgan Kaufmann Publishers.

В качестве примера возможностей ARM мы рассмотрим сравнительно новое расширение, которое компания ARM рекламирует под названием NEON.

Команды NEON появились в архитектуре ARMv7 – правда, не во всех ее реализациях. Процессоры ARM Cortex A8 поддерживают NEON в обязательном порядке, тогда как в процессорах ARM Cortex A9 эта поддержка необязательна. Чипсет Nvidia Tegra 2, который до сих пор используется в некоторых планшетах Android, не поддерживает NEON, а Nvidia Tegra 3 – поддерживает. NEON представляет собой набор команд SIMD, то есть инструкций, которые способны работать с несколькими единицами данных од-

«Концепция процессоров ARM изначально выглядела красиво.»

новременно. В отличие от реализаций SIMD в некоторых других архитектурах, для использования команд NEON процессор ARM не требуется переводить в особый режим. Инструкции NEON можно свободно смешивать с другими инструкциями ARM.

В распоряжении команд NEON 32 64-битных регистра, которые можно рассматривать также как 16 128-битных регистров. С точки зрения возможностей инструкции NEON не слишком отличаются от наборов инструкций SIMD в процессорах Intel. Это и неудивительно – ведь создавались они для решения одних и тех же задач.

Рассмотрим простой, самый простой пример. Пусть у нас есть канал стереозвука. Введем немного терминологии. Оцифрованный звук состоит из серии отсчетов [samples]. Поскольку стереозвук содержит два канала, минимальная единица звука – кадр [frame] состоит из двух отсчетов. В обычном стереопотоке отсчеты правого и левого каналов чередуются; таким образом, поток можно рассматривать как последовательность кадров. При обработке звука (например, фильтрации) бывает необходимо разде-

лить отсчеты разных каналов, обработать их, а потом восстановить чередование.

В системе команд NEON существуют команды загрузки чередующихся данных с автоматическим обращением чередования [deinterleaving] и сохранения данных с восстановлением чередования. Мнемонические обозначения этих команд имеют вид **vld*** и **vst***, где * – число 1, 2, 3 или 4, соответствующее порядку чередования данных. Это же число обозначает количество регистров NEON, которые должны использоваться в процессе обращения чередования. Например, при работе со стереозвук, где мы имеем два чередующихся канала, соответствующие команды должны выглядеть как **vld2** и **vst2**. Если бы нам нужно было обратиться чередование отсчетов интенсивности цвета в потоке формата RGBA, мы бы использовали команды **vld4** и **vst4**, при этом четыре задействованных регистра NEON содержали бы данные R, G, B и A соответственно.

На этом магия команд NEON не исчерпывается. Чередующиеся отсчеты могут иметь разную разрядность: 8, 16, 24 и 32 бита, и команды NEON позволяют учесть этот факт. Например, если нам нужно обратиться чередование стереопотока аудиоданных разрядностью 16 бит, команда загрузки данных будет выглядеть так: **vld2.16**. После имени команды следует список регистров NEON для загрузки данных. Число этих регистров должно соответствовать выбранной команде. Например, для стереозвуча эта команда может выглядеть так:

```
vld2.16 {d0, d1}, [r0]
```

Здесь d0 и d1 – регистры NEON, в которые будут загружены данные, а вместо r0 можно использовать любой обычный регистр ARM, который содержит адрес блока данных для загрузки.

Итак, в результате выполнения операции **vld2** в регистры d0 и d1 загружаются отсчеты считанных кадров, соответствующие правому и левому каналам. Сколько же всего кадров было считано? Ответ на этот вопрос позволяет определить, насколько эффективно использование NEON. Регистры d* являются 64-битными. Это значит, что в каждый регистр записано по 4 16-битных отсчета, то есть всего – 4 кадра. В принципе, в инструкции **vld*** допускается использование 128-битных регистровы q*; в этом случае количество данных, загруженных одной инструкцией, удваивается.

Напишем функцию swap_channels, которая меняет местами каналы стерео.

```
swap_channels:
vld2.16 {d0, d1}, [r0]
vst2.16 {d1, d0}, [r0]
mov pc, lr
```

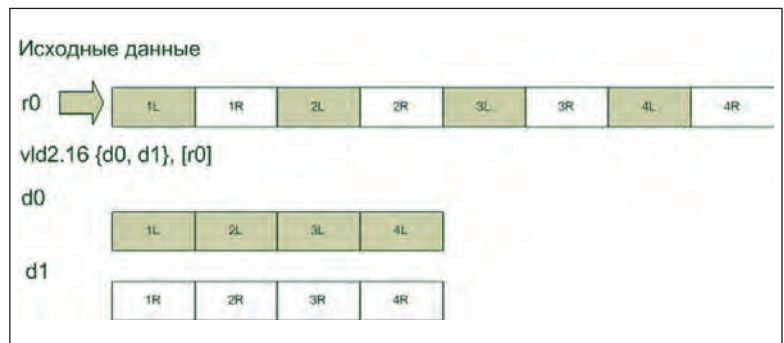
Эта функция, как вы догадались, написана на чистом ассемблере, и ее необходимо поместить в файл с расширением s. Согласно стандарту EABI, регистр r0 используется для передачи первого аргумента при вызове функции, так что из кода, написанного на C, ее следует вызывать так:

```
swap_channels(data);
```

где data – указатель на блок аудиоданных. Переставленные местами каналы записываются в ту же область памяти, где находились исходные.

Для сборки проекта с поддержкой NEON вам также понадобится внести изменения в файлы **Application.mk** и **Android.mk**. В первом файле переменной APP_ABI необходимо присвоить значение armeabi-v7a. Можно присвоить и сразу два значения: armeabi и armeabi-v7a. В этом случае ваш код будет скомпилирован в двух вариантах – с поддержкой NEON и без оной. Разумеется, это возможно только в том случае, если исходные тексты не содержат ассемблерных вставок, использующих инструкции NEON. Далее следует ввести переменную

```
LOCAL_ARM_MODE := NEON
```



И это еще не все. Допустим, что функция swap_channels сохранена в файле **swapchannels.s**. В таком случае в файле **Android.mk** в переменной LOCAL_SRC_FILES имя этого файла должно быть указано как **swapchannels.s.neon** (только в **Android.mk**, но не на диске!). Так же следует поступать и с другими модулями, в которых требуется поддержка NEON. Хотя необходимость добавлять окончания **neon** может показаться странной, на самом деле это весьма удобный способ отделить модули, в которых требуется поддержка NEON, от модулей, в которых она не нужна.

» Принцип работы команды vld2.

Root не нужен

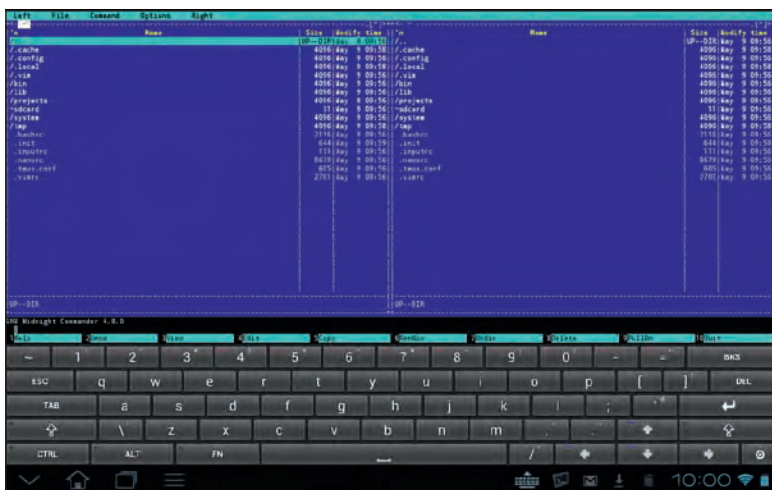
В прошлых статьях я писал о том, что для запуска программ Linux под управлением Android потребуется рутованное устройство или программный эмулятор. Но жизнь не стоит на месте, и я рад сообщить вам, что есть замечательная программа, которая позволит вам работать с устройством Android почти так же, как с любой машиной под управлением Linux, в том числе запускать программы для Linux, и все это – не требуя прав суперпользователя! Кроме того, данная программа относится к категории программ, способных разрабатывать приложения для Android на самом устройстве Android (и появление таких программ я тоже предсказывал!).

Речь идет о программе под названием *Terminal IDE*. На первый взгляд это название выглядит как «Уж и Еж», но не все так просто. Пакет Terminal IDE содержит весь необходимый инструментарий для сборки, отладки и запуска программ Java. Причем конечные файлы собираются не в формате APK, который требует установки, а в обычном для Java формате (правда, байт-код рассчитан на фирменную виртуальную машину Android). Впрочем, как мы с вами договорились, разработка программ Java в рамках этой серии нас не интересует, а интересует нас то, что все инструменты для работы с Java в Terminal IDE являются стандартными консольными программами (javac, java и т.д.), а, стало быть, пакету не обойтись без эмулятора терминала, который он и реализует (и делает это лучше, чем другие виденные мной программы).

»

Тест на сообразительность

Я знаю, о чем вы сейчас думаете. Вы думаете о том, что для перемены мест каналов стереозвуча достаточно сдвинуть поток на величину одного отсчета, а затем добавить недостающий отсчет в конце потока, и никакие специнструкции при этом не нужны. Я согласен. Но что поделать, если мне понадобился простой пример для работы с векторными инструкциями? В качестве оправдания отмечу, что в фирменной документации по NEON в качестве примера используется не намного более осмысленная операция перестановки компонентов в RGB-потоке.



► **Midnight Commander** в программе **Terminal IDE**.

Кстати, если уж речь зашла об именах: программа, написанная разработчиком, назвавшимся Spartacus Rex (имеется в виду вождь восставших рабов, а не футбольная команда), заслуживает внимания в силу одного этого факта.

Помимо эмулятора терминала, мы получаем в свое распоряжение оболочку *bash*, набор утилит *busybox*, набор программ для работы с *Git* и программу *Midnight Commander*. Кроме того, пакет обладает замечательной программистской клавиатурой (она заслуживает отдельного описания), возможностью подключаться к другим системам с помощью программ *telnet* и *ssh*, а также возможностью подключаться к устройству Android с помощью тех же *telnet* и *ssh*. В последнем случае придется использовать так называемый обратный туннель SSH – режим, в котором клиент SSH, запущенный на компьютере, может предоставлять доступ другим системам через SSH. Помимо, все вышеперечисленное уже звучит достаточно заманчиво для того, чтобы поспешить с установкой *Terminal IDE* в свою систему. Но главное сокровище *Terminal IDE* для нас – это собственная файловая система пакета, которая дает нам полный контроль над файлами, в том числе позволяет присваивать им статус исполняемых. При этом разделяемые библиотеки Android остаются видимыми для программ, работающих в этой файловой системе.

Между прочим, в руководстве пользователя *Terminal IDE*, которое автор программы советует прочесть не менее двух раз, прежде чем задавать вопросы ему, написано, что выполнять динамически

► Программа **threads** в окне **Terminal IDE**.



скомпонованные программы Linux из терминала нельзя, так как динамический загрузчик Android отличается от стандартного загрузчика Linux. Мы уже знаем, как обойти данное ограничение, так что это предостережение не для нас (в качестве демонстрации запустим в *Terminal IDE* скомпонованную динамически программу *threads*, исходники которой вы найдете на диске).

Пройдя процедуру установки программы и ее настройки через стартовый экран (кстати, самые интересные опции настроек доступны в разделе настроек на стартовом экране, а не в обычном разделе настроек приложения Android), вы сможете запускать терминал (программа позволяет открыть несколько окон терминала, между которыми переключаются, «пролистывая» их на сенсорном экране). Среди набора доступных нам программ, к сожалению, нет компилятора GCC, ведь, как уже было сказано *Terminal IDE* предназначена для разработки программ на Java. Однако ничто не мешает нам использовать терминал для запуска программ, собранных на ПК с помощью средств ARM GCC.

Копируем исполняемый файл такой программы в одну из директорий нашего устройства, и с помощью *Terminal IDE* пытаемся присвоить файлу атрибут исполняемого. Ничего не происходит.

Причина в том, что мы работаем не там, где надо. Чтобы получить власть над файлом программы, нам надо скопировать или переместить его в собственную файловую систему *Terminal IDE*. В окне тер-

минала вводим следующую последовательность команд:

```
cp program ~
cd ~
```

Клавиатура для программистов

Как известно, виртуальная клавиатура Android разрабатывалась с расчетом на пользователей, а не на программистов. Кроме того, при разработке этой клавиатуры учитывался тот факт, что пользователь имеет дело с сенсорным экраном и с приложениями, наделенными графическим интерфейсом. Именно поэтому на виртуальной клавиатуре Android отсутствуют клавиши стрелок, клавиша Esc (ее функции иногда берет на себя клавиша «назад», расположенная за пределами виртуальной клавиатуры), клавиши Ctrl и Alt, а также клавиши F1–F12. Во время как эти клавиши совершенно не нужны для ввода подписей к фоткам, смайликов или почтовых адресов, пользователя, работающим со стандартными инструментами Linux, без этих клавиш очень трудно обойтись.

Виртуальная клавиатура *Terminal IDE* продумана как в функциональном плане, так и в отношении эргономики. Она не только обладает всеми перечисленными клавишами, но и учитывает специфику работы с мобильным устройством. Клавиши Ctrl, Alt и Shift являются залипающими, что совершенно необходимо, когда одной рукой вы держите устройство, а другой вводите текст. «Залипшие» в данный момент клавиши подсвечиваются, так что вы не ошибетесь при вводе.

Собственно, недостатков у этой клавиатуры я обнаружил только два. Один из них является принципиально неустранимым; второй, я надеюсь, будет устранен. Первый недостаток виртуальной клавиатуры *Terminal IDE* связан с тем, что, поскольку клавиш на ней больше, чем на стандартной клавиатуре Android, размеры самих этих клавиш (при равных размерах клавиатур) получаются меньшими. Вторая проблема, с которой столкнутся жители не-англоязычных стран – отсутствие поддержки любых языков, кроме английского. Надеюсь, что в будущем этот недостаток исправят.



► Клавиатура *Terminal IDE* в режиме отображения функциональных клавиш.

```
chmod 777 program
./program
```

Здесь `program` – исполняемый файл программы Linux.

Для расположения собственной файловой системы *Terminal IDE* использует директорию `/data/data/com.spartacusrex.spartacuseide/files`, то есть одну из тех директорий, к которым на нерутованном устройстве получить доступ с помощью обычного терминала нельзя. *Terminal IDE* получает доступ к этой директории как «честное» приложение Android. Таким образом, мы снова используем возможности приложений Android для того, чтобы расширить возможности приложений Linux.

С точки зрения самой программы *Terminal IDE* указанная директория представляется как домашняя директория пользователя (именно на нее указывает тильда при работе в `bash`). Внутри этой директории вы неожиданно найдете поддиректории `bin` и `lib`. Директория `bin` содержит все программы, которыми пользуется *Terminal IDE* (программы скомпилированы статически), а директория `lib` ничего не содержит.

В заключение отмечу, что программа весьма требовательна к объемам памяти. В хранилище приложение занимает 104 мегабайта (в моей системе это рекорд). В оперативной памяти занятый объем колеблется в районе 20 мегабайт. В этот объем, разумеется, не входят программы, запущенные из терминала, которые вообще невидимы для системы. Так что при работе с *Terminal IDE* на устройствах с небольшим объемом оперативной памяти следует соблюдать осторожность. *Terminal IDE* отнюдь не случайно является одной из тех программ, которые используют строку состояния Android для предупреждения пользователя о том, что они запущены. Кроме того, программа допускает явное завершение своей работы, а не перекладывает этот процесс на плечи Android.

Если хакеров можно в каком-то смысле сравнить с рыцарями, то курс молодого рыцаря на этом закончен. Вы владеете всем необходимым, чтобы продолжить самостоятельное изучение тонкостей механики ARM и Android для расширения возможностей этой ОС. **LXF**



Linux

Mandriva Académie

Операционная система Linux

Академическая программа для учебных заведений

Mandriva Linux

Mandriva.Ru предоставляет учебным заведениям лицензию, дающую право на неограниченное по числу рабочих станций использование дистрибутива Mandriva Linux на всех компьютерах в образовательном учреждении, всех компьютерах преподавателей и всех компьютерах учащихся, в том числе и домашних.

По этой программе учебное заведение получает:

- ★ свежие версии дистрибутива Mandriva Linux (дважды в год)
- ★ доступ к обновлениям системы
- ★ техническую поддержку

Комплект поставки:

Mandriva Linux Powerpack 2009.1 Spring — 32- и 64-битные версии (2 DVD), а также печатное руководство

- ★ Mandriva Free 2009.1 Spring
- ★ Mandriva One 2009.1 Spring
- ★ Репозиторий Mandriva 2009 — бинарные пакеты для платформы x86 (4 DVD)
- ★ EduMandriva (1 DVD) — дополнительное ПО для образования
- ★ Академическая лицензия

www.mandriva.ru

Тел.: (812) 309-06-86, (499) 271-49-55
info@mandriva.ru

Сравнение

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

Альтернативные операционные системы

Linux — не единственная операционная система с открытым кодом. Маянк Шарма поможет вам выбрать удобную альтернативу.



Про наш тест...

Поскольку тестировалось удобство в использовании, то, вопреки собственным рекомендациям, мы запускали эти операционные системы на настоящем оборудовании, а заодно и на виртуальных машинах.

В первую очередь нас интересовало, можно ли использовать эти ОС для повседневных задач. Экзотическое оборудование мы не брали, но пытались соединиться с Интернет через беспроводную карту и запускали ОС на одноядерных и многоядерных Pentium без встроенной видеокарты.

ОС сравнивались по установленным по умолчанию приложениям и предоставлению пользователям возможности добавлять пакеты. Поскольку мы ищем разумные альтернативы, ОС, способные работать со знакомыми многим из вас приложениями с открытым кодом, оценивались выше. Мы также оценивали их программы установки и опции поддержки.

Наша подборка

- » GhostBSD
- » Haiku
- » Icaros Desktop
- » OpenIndiana
- » PC-BSD

Мир открытого кода завален альтернативными операционными системами. Разработанные энтузиастами, многие из них и близко не подходят к популярности Linux в плане общего удобства в использовании. Ну и зачем бы вам использовать альтернативную ОС? Помимо того, что работа с такой повысит ваши профессиональные навыки, некоторые из этих непривычных дистрибутивов вполне со-

зрели для использования в качестве постоянной настольной системы. Так что мы решили предоставить вам возможность выбрать те, что пригодны для обычных повседневных задач — работа в сети, прослушивание музыки, просмотр DVD, игры и чтение PDF. Хотя некоторые из рассмотренных здесь ОС существуют уже более десяти лет, их разработчики по-прежнему позиционируют их как релизы ранней стадии разработки. Но это скорее отражает

не меру удобства ОС, а высокие стандарты разработчиков. Еще одна удивительная сторона этих не-Linux операционных систем — их сильное сообщество пользователей и разработчиков. Фактически, один из наших конкурсантов недавно помог другому улучшить работу с USB. И хотя сейчас альтернативных ОС меньше, чем пять лет назад — с учетом темпов роста современных ОС некоторые, похоже, станут мейнстримом уже в ближайшем будущем.

Приложения по умолчанию

Заработают ли они сразу?

Тот факт, что операционные системы не принадлежат к основным и большинство разработчиков программ не делает для этих платформ пакетов своих приложений, не означает, что они непригодны для повседневных задач.

В PC-BSD имеются такие популярные приложения, как медиа-плееры *Mplayer*, *Amarok* и *Dragon Player*. Для офисной работы предусмотрен пакет приложений *Koffice* вместо *LibreOffice*. Также имеется *Ktorrent* для скачивания торрентов, менеджер загрузок *Kget*, *Kopete* для чата и *Konqueror*, который одновременно служит web-браузером по умолчанию.

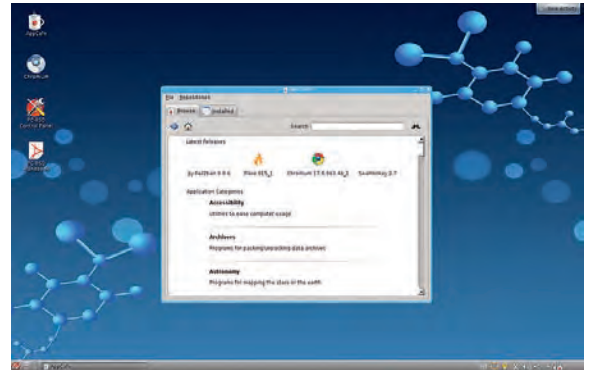
По умолчанию рабочим столом PC-BSD является KDE, но можно установить Gnome, Xfce или LXDE. OpenIndiana использует рабочий стол Gnome с *Compiz-Fusion* для внешней привлекательности. ОС использует старый релиз *Firefox* с чат-клиентом *Pidgin*, *Thunderbird* для электронной почты и *Totem* и *Rhythmbox* для воспроизведения мультимедиа. Единственное имеющееся здесь офисное приложение – это программа для чтения документов *Evince*. В GhostBSD также используется классический ра-

бочий стол Gnome с недавним релизом *Firefox*, чат-клиентом *Empathy*, медиа-плеером *Totem*, *Evolution* для электронной почты и *Transmission* для торрентов. Это также единственная ОС в нашем Сравнении, которая по умолчанию включает мощный офисный пакет *LibreOffice*.

Преимуществом этих трех ОС является использование на рабочих столах ряда популярных инструментов с открытым кодом, с которыми пользователи Linux хорошо знакомы. Благодаря этим знакомым программам, вы не заметите разницы между этими ОС и Linux, пока не копнете глубже.

В Icaros или Haiku вы столь знакомой атмосферы не найдете – обе применяют собственные настраиваемые приложения; и в обеих этих приложениях тонны. Всё, включая web-браузеры, клиенты IRC, редакторы изображений, приложения для электронной почты и управления контактами, медиа-плееры, программы для просмотра изображений и PDF, продвинутые текстовые редакторы, инструменты для воспроизведения музыки и утилиты деления диска на разделы, здесь имеется.

Однако некоторые из приложений Icaros менее функциональны, чем у Haiku. Например, браузер



» Icaros Desktop – лучшая ОС для фанатов игровой классики.

WebPositive в Haiku можно использовать с модулем расширения *gnash* для отображения Flash, а браузер *Origyn* в Icaros этому трюку не обучен.

Но главное достоинство Icaros – его верность наследию Amiga: он является отличной игровой платформой. В Icaros множество игр, включая shareware-версии *Doom* и *Quake*, и множество игр на базе библиотеки SDL, таких, как *Blob Wars* и *X-Penguin*. Здесь также есть эмуляторы для классических игровых платформ.

Несмотря на их сходство, мы полагаем, что из них двоих Icaros более эзотерический. Без руководства пользователя начинающие пользователи вряд ли даже найдут большинство его приложений.

«Главное достоинство Icaros – его верность наследию Amiga.»

Вердикт

Haiku
★★★★★
Icaros Desktop
★★★★★
PC-BSD
★★★★★
GhostBSD
★★★★★
OpenIndiana
★★★★★

» Haiku и Icaros наиболее полные.

Кодеки и модули расширения

Справляются ли эти операционные системы с мультимедиа?

РC-BSD идет с модулем расширения Flash Player, который работает с *Konqueror* и другими web-браузерами, загружаемыми из его репозитория. Но если вам нужен Java, придется искать Java-

браузер *IcedTea* в репозиториях. PC-BSD также устанавливает несколько медиа-плееров, в том числе *Mplayer* и *Amarok*, снабженных кодеками для воспроизведения всех форматов медиа.

Напротив, браузер *Firefox* в GhostBSD не включает модуль расширения Flash, как и весь дистрибутив не включает модулей для воспроизведения файлов в проприетарных форматах. Однако вы можете установить модуль для Flash и несколько других через коллекцию портов BSD.

OpenIndiana тоже не содержит модулей расширения, но вы можете скачать такие полнофункциональные плееры, как *VLC*

и *Mplayer*, из репозитория SFE. Модуль Flash вам придется скачать с сайта Adobe.

По части воспроизведения мультимедиа и у Haiku, и у Icaros свои причуды. Чтобы настраиваемый медиа-плеер Haiku воспроизводил мультимедиа, вам придется заняться кунг-фу с CLI. А ее браузер *WebPositive* будет воспроизводить Flash, только когда вы приставите к нему *gnash*.

Звук в Icaros отключен по умолчанию, но включается без особых хлопот. ОС содержит *Mplayer* и умеет работать с разнообразными медиа-форматами. Ее браузер *OWB* не имеет поддержки Flash, но для воспроизведения видео с YouTube имеется небольшой скрипт, который берет URL в *OWB* и передает его на *Mplayer* для воспроизведения.



» Не все функции в Icaros работают с live CD.

Вердикт

PC-BSD
★★★★★
GhostBSD
★★★★★
OpenIndiana
★★★★★
Icaros Desktop
★★★★★
Haiku
★★★★★

» При потребности в мультимедиа избегайте Haiku и Icaros.

Удобство и производительность

Насколько они отличаются от вашего обычного рабочего стола Linux?

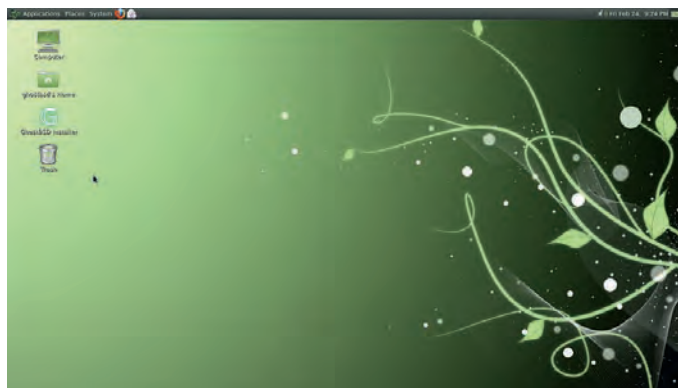
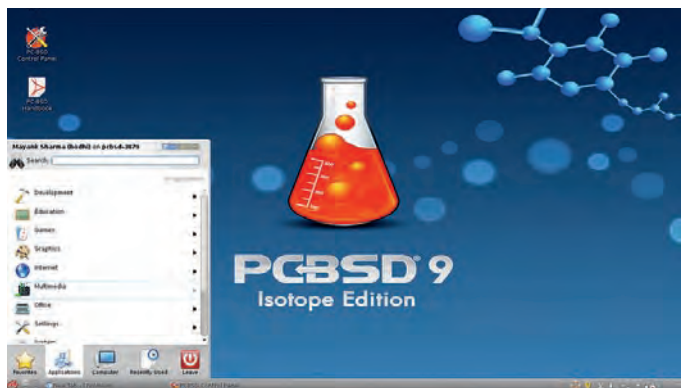
Альтернативные операционные системы в нашем Сравнении рассматриваются как таковые не без причины – они не похожи на ваш повседневный дистрибутив Linux. Было бы неразумно ожидать, что они будут выглядеть, ощущаться и вести себя, как другие популярные ОС. Но при всем при том, так ли сильно они отличаются от вашего дистрибутива Linux? Возможно, изнутри – да, но снаружи у них весьма похожие рабочие

столы, а то и вообще KDE или Gnome. Каждая ОС в Сравнении пришла в жизнь по определенной причине, и каждая имеет свои фишки. Одни малы, отличаются легким весом и работают со скоростью света, а другие предлагают обширную подборку приложений. Однако – если на миг забыть о том факте, что большинство из них пока на ранних стадиях разработки – есть ли в них то, что позволит вам использовать их для повседневных надобностей?

PC-BSD ★★★★★

С места в карьер скажем, что работа с PC-BSD очень сильно похожа на работу с дистрибутивом Linux – в основном потому, что она использует те же приложения с открытым кодом, что сидят на вашем привычном рабочем столе.

Но в ней есть несколько настраиваемых инструментов, которые делают ее очень дружелюбной к пользователю. Control Center помогает управлять разными аспектами системы – например, добавлением новых пользователей и настройкой сетевых соединений. Затем, здесь имеется *Life Preserver*, инструмент резервного копирования, способный безопасно синхронизироваться с удаленной системой FreeNAS через *rsync* и SSH. Нам пришлось отключить WEP, чтобы заставить одну сетевую карту Linksys соединиться, но с остальными проблем не было. Также нам пришлось повозиться с инструментом настройки X, чтобы добиться нужного разрешения экрана для карты Nvidia.



GhostBSD ★★★★★

GhostBSD превосходит PC-BSD только наличием у нее live CD. Она тоже не слишком отличается от дистрибутива Linux, благодаря рабочему столу Gnome. Однако по части оборудования нам удалось заставить GhostBSD работать без проблем только на одной из наших машин. На других или внезапно во время установки замерзала клавиатура, или возникали проблемы с выбором беспроводной карты, как и с проводным сетевым адаптером, а это уж странно.

Также у нас были проблемы с менеджером пакетов, который очень медленно работал с обновлением из репозитория, и наблюдались графические несоответствия. Например, изначально система сообщала, что *Firefox* не был установлен, но при нажатии на кнопку установки заявляла, что приложение уже есть. Также при поиске пакета *VLC* произошел сбой.

Вдохновение

Почему возникли эти операционные системы?

Подобно большинству дистрибутивов Linux, эти ОС родились благодаря любопытству. И хотя появились они как любительские, было бы несправедливо по-прежнему относить их к этому разряду, принимая во внимание, что некоторым из них уже более десяти лет.

И PC-BSD, и GhostBSD основаны на FreeBSD и направлены на то, чтобы донести прелесть ОС уровня предприятия до пользователя рабочего стола. Проект PC-BSD начался в 2005 г. с графической

программы установки и инструмента настройки для рабочего стола KDE, и сейчас включает собственную настраиваемую систему управления пакетами PBI. GhostBSD пытается сделать нечто похожее для пользователей Gnome. Однако, в отличие от PC-BSD, она доступна в устанавливаемой среде live. OpenIndiana появилась как ответвление OpenSolaris, после того, как Oracle прекратил разработку последней. Проект работает благодаря Illumos Foundation и обладает всем для достижения своей цели стать «фактическим дистрибути-

вом OpenSolaris». Подобным же образом, Haiku появилась, когда заглохла BeOS. Она существует с 2001 г., но у нее было всего три релиза, хотя проект обещает быть совместимым с BeOS по исходникам и бинарникам. Она все еще находится в стадии активной разработки, и ее разработчики утверждают, что она уже способна на большее, чем BeOS. Icaros Desktop приносит на рабочий стол исследовательскую ОС AROS с открытым кодом. Хотя AROS – это реализация AmigaOS, ОС не является двоично-совместимой с AmigaOS.

Вердикт

GhostBSD	★★★★★
Haiku	★★★★★
Icaros Desktop	★★★★★
OpenIndiana	★★★★★
PC-BSD	★★★★★

» Все ОС получают 5/5 за старания и упорство.

OpenIndiana ★★★★★

Тему сходства продолжает OpenIndiana, где используется более старый релиз классического рабочего стола Gnome. Кстати о старом ПО: версия браузера Firefox здесь доисторическая, 3.6.12.

Новые пользователи оценят ее графическую программу установки. В этой ОС также имеется утилита Device Driver, которая поможет выяснить, нужны ли дополнительные драйверы вашему оборудованию. На одном компьютере она не смогла загрузить драйвер Intel Ethernet, но беспроводная карта Linksys работала по умолчанию. В ней есть собственный графический сетевой менеджер, но для настройки сети нужно привыкнуть и освоить жаргон ОС. Здесь также имеется инструмент для дотошной настройки брандмауэра и утилита Time Slider – она делает резервные копии данных, создавая моментальные снимки файловой системы ZFS через заданные промежутки времени.



Haiku ★★★★★

Haiku мы немного опасались, учитывая ее стадию релиза alpha 3. Но удобство ее использования нас просто ошарашило. Интерфейс у ОС прост, но не следует считать его примитивным.

Haiku – на редкость гибко настраиваемая операционная система, и в ней есть ряд знакомых приложений, способных справиться со всеми типами файлов, которые вы им скормите. По мере углубления в использование ОС и появления проблем вы также по достоинству оцените очень активное сообщество разработчиков и пользователей.

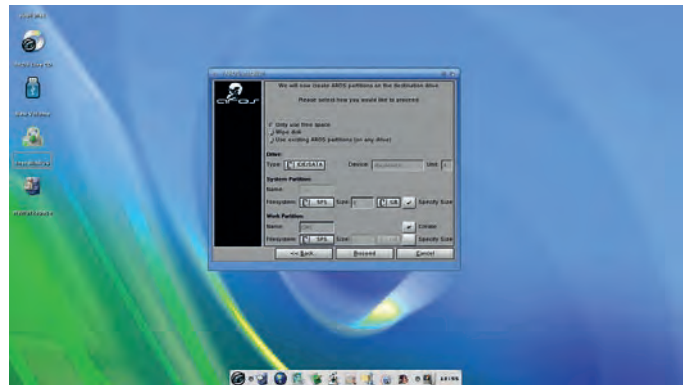
Haiku – многопоточная операционная система, и один из наиболее интересных ее инструментов – менеджер потоков. Он показывает, сколько памяти потребляет каждый поток, и позволяет вам убить поток одним щелчком мыши.

Icaros Desktop ★★☆☆☆☆

По необычности с Icaros Desktop другим ОС не сравнятся. Если вы незнакомы с AmigaOS былых времен, осваивать Icaros будет туго.

Установка запутанная, и тяжело даже подступиться к освоению ОС, не прочитав сперва ее руководство пользователя. Начинающие пользователи оценят тот факт, что ОС распространяется в виде устанавливаемого live CD, чтобы разжечь их аппетит.

В Icaros весьма достойная поддержка оборудования, но все же не ждите, что она станет работать со сверхнавороченными видеокартами и со всем беспроводным сетевым оборудованием. Так же, как и Haiku, Icaros имеет все необходимые для постоянной настольной системы приложения, но ее истинным коньком являются ретро-игры, и именно в этой области она раскрывается во всем блеске.



Поддержка

Не приступить к работе? Не беда. Помощь есть прямо на рабочем столе.

Документация и поддержка важны для любого проекта, но их роль еще возрастает, когда речь идет об операционных системах вне мейнстрима. PC-BSD добивается неплохого результата. В ней есть wiki с информацией по всем видам совместимого оборудования, множество руководств в базе знаний в стиле FAQ и справочник пользователя, встроенный в установленную систему. Есть также активный форум, где можно попросить PBI. В Haiku тоже учли все ваши интересы и основные пункты. Здесь есть ряд само-

достаточных руководств, чтобы помочь пользователю разобраться с определенными задачами, такими, как установка Haiku и программ, и руководство пользователя все-в-одном, которое также встроено в ОС. Чтобы вы получили представление об ОС, у них есть профессионально снятое видеоруководство. Есть также активный форум для поддержки пользователей.

GhostBSD содержит кучу видео, поясняющих наиболее общие задачи, вроде установки и перенастройки пароля root. На их форумах вы найдете советы и HowTo,

и на подходе справочник пользователя. В OpenIndiana имеется wiki, рассматривающая наиболее частые действия, например, установку пакетов, и поддерживаемый сообществом список совместимости оборудования. А вот форума пока нет, так что по вопросам поддержки придется обращаться в списки рассылки.

В Icaros Desktop без документации никак не обойтись, и она вас не разочарует своим всесторонним руководством пользователя, также включенным в ОС. Подсказки рассредоточены по активному форуму.

Вердикт

- Haiku ★★★★★
- PC-BSD ★★★★★
- GhostBSD ★★★★★
- Icaros Desktop ★★★★★
- OpenIndiana ★★★★★

» Встроенная документация – во главе всего.

Доступность приложений

Позволяют ли они управлять пакетами?

Путь приложений и много – а ну как вы решите установить еще? Все ОС в нашем Сравнении позволяют установить добавочные приложения, но не во всех это требует одинаковых сил. У Icaros и Haiku нет менеджеров пакетов, но их сообщества уверяют, что и недостатка в приложениях нет. У обеих ОС имеются сайты с дополнительными пакетами. Благодаря их архитектуре, установка обычно сводится к скачиванию пакета и распаковыванию его содержимого в директорию.

При наличии всех обычных приложений рабочего стола, в Icaros имеется больше игр и инструментов разработчика. Haiku предлагает более широкую палитру. Так,

на Haikuware.com, помимо плееров видео и аудио и творческих приложений, приложений для работы, типа текстовых редакторов и утилит для записи DVD, есть также драйверы для всех видов оборудования.

OpenIndiana щеголяет *Synaptic*-подобным приложением для добавления пакетов – Package Manager. По умолчанию он настроен на поиск и скачивание пакетов из официального репозитория openindiana.org. Помимо постоянного издателя, есть два других официальных репозитория. Репозиторий экспериментальных программ содержит пакеты, находящиеся в разработке. В сторонних источниках пакетов – например, репозитории Spec File Extra –

имеются также программы, обремененные патентами. И GhostBSD, и PC-BSD пользуются коллекцией портов и пакетов FreeBSD, насчитывающей 20000 пакетов. В GhostBSD можно установить эти пакеты, не прибегая к командной строке, через графический интерфейс.

Нажмите кнопку

PC-BSD также публикует пакеты в собственном формате файлов для программы установки по кнопке (push-button installer, PBI), их можно установить через менеджер пакетов *AppCafe*. Файл PBI самодостаточен и включает все зависимости, библиотечные и времени выполнения [runtime], нужные приложению. Он создан из исходного пакета FreeBSD и разработан с целью избежать установки многократных копий одной и той же библиотеки. Во время установки система PBI сравнивает установленные библиотеки и файлы с содержащимися в файле PBI и устанавливает только те, которых еще нет в системе. *AppCafe* отличается быстротой и дружелюбием к пользователю. Его главный интерфейс отображает самые свежие релизы и предоставляет опцию поиска конкретных приложений. Он также позволяет узнать о появлении новых версий для установленных PBI. Его единственный мелкий недостаток – то, что он не показывает размер приложения, пока не начнется его загрузка.



➤ PC-BSD имеет онлайн-инструмент обновления, привязанный к *AppCafe*.

Вердикт

- GhostBSD ★★★★★
- PC-BSD ★★★★★
- OpenIndiana ★★★★★
- Haiku ★★★★★
- Icaros Desktop ★★★★★

» Система PBI PC-BSD выделяет ее в группе.

Установка ОС

Требуются ли особые навыки?

У всех наших ОС имеется графическая программа установки. Если вы помните об обычных мерах предосторожности и будете следовать инструкциям разработчиков, у вас не должно возникнуть никаких проблем при их установке на настоящий жесткий диск.

OpenIndiana идет с отлаженной и быстрой программой установки, и у вас не возникнет с ней проблем, если вы уже устанавливали дистрибутивы Linux. Здесь есть *Gparted* для подготовки диска, хотя диск можно разбить на разделы и из самой программы установки. Несколько раздражает разве что невозможность высказать свое мнение по поводу установки программы загрузки, которая устанавливается автоматически на MBR. В программе установки

HaikuOS тоже можно делить диск на разделы, и по умолчанию загрузочный сектор пишется в собственный раздел. Программа установки проста и понятна в навигации, и производит установку быстрее всех прочих ОС.

Оба потомка BSD имеют похожие программы установки, но установщик PC-BSD производит впечатление более продуманного. Обе программы умеют разбивать диск на разделы. Серьезная проблема программы установки GhostBSD – то, что на некоторых компьютерах ее нельзя использовать: в среде live не работает клавиатура, и вам придется использовать текстовый установщик. Но самой хитроумной в установке безусловно является Icaros, особенно если вы хотите, чтобы она де-

лила диск с другими ОС. Самый простой и безопасный способ установить Icaros – позволить ей занять весь диск целиком. И даже тогда мы не рекомендуем приступать к установке ОС, не прочитав руководство пользователя.



➤ Помните, вы не можете установить BSD на раздел extended.

Вердикт

- Haiku ★★★★★
- PC-BSD ★★★★★
- GhostBSD ★★★★★
- OpenIndiana ★★★★★
- Icaros Desktop ★★★★★

» Лучше всего использовать сторонний загрузчик.

Альтернативные ОС

Вердикт

Выбирать победителя в данном конкретном Сравнении будет несправедливо: это все равно что сравнивать яблоки с апельсинами. Единственное, что есть общего у этих операционных систем – то, что они не Linux. Кроме того, одни находятся в ранней стадии разработки уже лет десять, другие обладают нечестным преимуществом, поскольку опираются на уже существующие крайне популярные проекты, а остальным приходится развиваться с нуля.

Работа с любой (или со всеми) ОС безусловно добавит вам профессионализма. Но поскольку намерением нашего Сравнения было дать вам настольную операционную систему, пригодную для повседневной работы, мы выбрали победителя, основываясь на его пользе и удобстве для большинства пользователей.

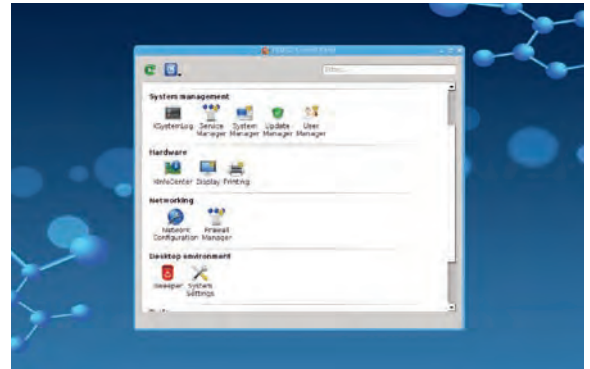
OpenIndiana, Haiku и Icaros Desktop лучше в том, что они делают. Смысл их существования в том, чтобы донести до пользователя рабочего стола все преимущества их предшественников – Solaris, BeOS и Ami-

gaOS соответственно, и все они достойны похвалы. Хотя OpenIndiana довольно нова, она проще всех для неопытных пользователей, благодаря знакомому интерфейсу и настраиваемым инструментам, разработанным в помощь начинающим.

Зато Haiku подойдет пользователям, желающим получить ОС, не слишком требовательную к оборудованию и снабженную всеми приложениями, необходимыми настольной системе. Сложите эти два условия вместе – и у вас получится отличная ОС для того, чтобы вдохнуть жизнь в относительно старое железо.

Из всех операционных систем Icaros Desktop показалась нам наименее удобной для среднестатистического пользователя, и весьма сложной в освоении для непривычных к стилю AmigaOS. Иными словами, это идеальная ОС для вас, если вы – фанат ретро-игр (а кто не фанат?).

Итак, остаются отпрыски BSD. Максимальному числу пользователей настольных сис-



тем будет полезна PC-BSD. Можно сказать, что PC-BSD для FreeBSD является тем же, чем Ubuntu для Debian. Хотя по умолчанию в ней используется KDE, последняя версия также содержит другие рабочие столы, в том числе Gnome и Xfce.

В PC-BSD есть все инструменты, требуемые обычной настольной системой. По сути, это идеальная замена вашему дистрибутиву Linux практически для любой задачи.

► Спонсор PC-BSD – iXsystems, который также поддерживает мощный сервер с открытым кодом FreeNAS.

«По сути, PC-BSD – идеальная замена вашему дистрибутиву Linux.»

I PC-BSD ★★★★★

Сайт: www.PC-BSD.org Лицензия: Различные лицензии BSD
 » Нужна настольная ОС – не Linux, но легкая в освоении? Вот она.

IV GhostBSD ★★★★★

Сайт: www.ghostbsd.org Лицензия: BSD
 » Путь пользователей Gnome к FreeBSD.

II OpenIndiana ★★★★★

Сайт: www.openindiana.org Лицензия: CDDL и другие лицензии
 » OpenSolaris живет!

V Icaros Desktop ★★☆☆☆☆

Сайт: www.icarosdesktop.org Лицензия: AROS Public License
 » Лучшая ОС для штурма продуктивности.

III Haiku ★★★★★

Сайт: www.haiku-os.org Лицензия: MIT и другие лицензии
 » Лучшая альтернативная ОС, о которой вы не слышали.

Обратная связь

Вы используете одну из этих ОС в качестве повседневного рабочего стола? Расскажите нам об этом: ixf.letters@futurenet.co.uk.

Рассмотрите также...

Нетривиальная цель ReactOS добиться совместимости с программами и драйверами Windows XP наконец начала срастаться. Теперь у пуристов свободного ПО появилось несколько опций для работы вместе с ядром GNU/Hurd; помимо Debian GNU/Hurd, есть основанный на Arch live CDи неофициальный порт Gentoo.

Затем есть целевые ОС с серией дополнений. Одна из них – FreeVMS, она работает на 64-битной архитектуре и во многом

подобна LoseThos – что будет интересно студентам, изучающим компьютерную технологию. Есть несколько ОС с ограниченным набором приложений типа Visopsys; Jnode, чья цель – работа с любыми приложениями Java; Syllable – клон AmigaOS; и KolibriOS, ответвление от другой альтернативной ОС, MenuetOS. Есть также несколько операционных систем с закрытым кодом, например, Menuet 64 и SkyOS. **ixf**

Надуем



World Wide Web — величайшая из когда-либо созданных систем для предоставления информации, но как пресечь злоупотребление доступом? Исследует **Бен Зверард**.

У вас не параноя: за вами реально следят. Преступные элементы, web-компании и правительственные службы — у них у всех есть

причины шпионить за вашей жизнью онлайн, и применяемые ими методы становятся все изощреннее.

2011 год был годом испытаний для обитателей Сети, в особенности несогласных с тем, что творит их правительство. По всему миру 199 человек было арестовано или задержано из-за их публикаций онлайн. Многие из этих людей до сих пор томятся в тюрьмах заключения.

Вызвавшая санкции информация варьировалась от разоблачения вреда окружающей среде до религиозных поучений и критики властей.

«199 человек было задержано из-за их публикаций онлайн.»

Вдобавок происходит увеличение использования информации об обитателях сети, которой владеют web-компании. Политики приватности расширены, и сейчас Твиттер продает права

на данные пользователей. Некоторые из представленных ниже методов самозащиты могут повлиять на ваше использование компьютера, хотя для большинства людей реализация всех этих методов была бы чрезмерной.

Наша цель состоит в демонстрации, что можно разузнать о вас и кто может это сделать, а также как это предотвратить. Что делать с полученной информацией, в общем-то, решать вам. Беспокоит ли вас объем информации, собираемой компаниями, скрываетесь ли вы от коррумпированных властей — читайте, и вы узнаете, как сохранить свои данные только для себя.



Узнать, сколько именно информации вы открываете внешнему миру, вам поможет *Wireshark*. Эта программа анализирует информацию, проходящую через сетевые интерфейсы, и позволяет выполнять в ней поиск и фильтрацию по заданным шаблонам. Она берет информацию с вашего сетевого интерфейса, то есть все отображаемые ею данные являются видимыми всем (потенциально злонамеренным) пользователям в сети.

Wireshark должна быть доступной в вашем менеджере пакетов, или на wireshark.org. После установки запускается она так:

```
sudo wireshark
```

Вы получите сообщение, что программа запущена с привилегиями суперпользователя и что так поступать нехорошо. Если вы планируете часто пользоваться этой утилитой, последуйте руководству по правильной настройке, однако на первый раз это предупреждение можно про-

пустить. Щелкните по сетевому устройству в списке интерфейсов (для проводного соединения это, скорее всего, eth0, а для беспроводного — wlan0), и начнется захват данных. Когда вы войдете в сеть, верхняя часть экрана начнет заполняться разноцветными пакетами. Утилита содержит фильтр, чтобы было легче разобраться в этой пестрой смеси. Например, можно полюбопытствовать поиском в duckduckgo.com с помощью фильтра:

Обновляйтесь

Рассматривайте свой web-браузер и библиотеки SSL как критические приложения. Чтобы никакие злонамеренные типы не смогли воспользоваться их ошибками для обхода шифрования, постоянно применяйте к ним заплатки безопасности.

вы посылаете; а вот если вы выберете протокол справа, данные увидит только их адресат.

Для работы в Интернете важна последняя строка таблицы. Как мы видели ранее, многие компьютеры могут прочитать то, что послано по HTTP, однако если мы снова проделаем наш тест, используя безопасную web-страницу <https://www.duckduckgo.com> (обратите внимание на s), вы увидите, что в *Wireshark* информация уже не появится.

В некоторых web-браузерах при соединении с безопасной страницей отображается висячий замок, однако вместо него легко подsunуть значок web-страницы (см. рисунок слева). Если вы не уверены, щелкните по значку. На настоящем замке откроется диалог с информацией о безопасности этой страницы.

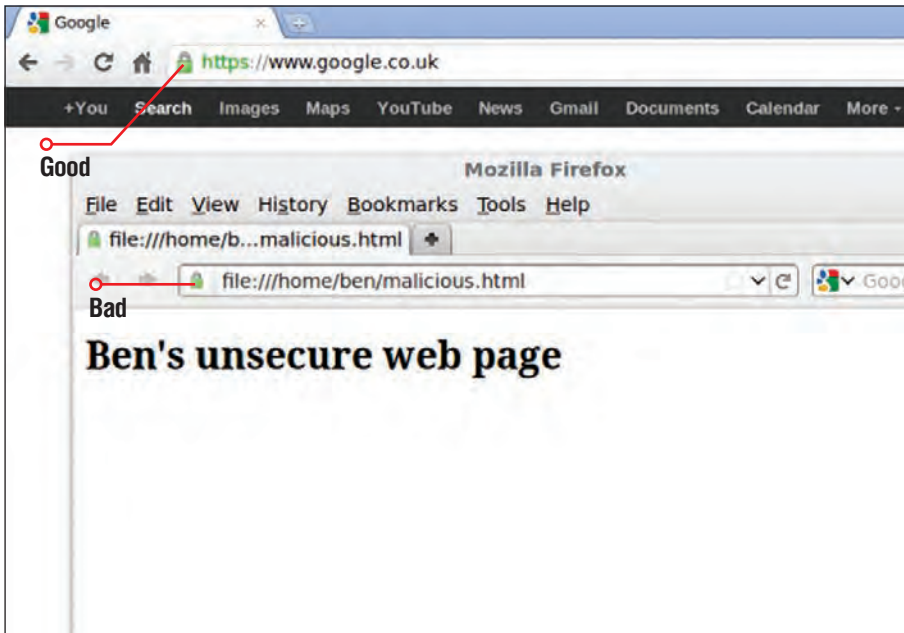
Конечно, это гарантирует лишь то, что информация не будет прочитана при передаче с вашего компьютера на сервер. Но организация, владеющая сервером, может передавать ее третьим лицам или переправлять небезопасным образом между своими дата-центрами. Отправив информацию,

вы теряете контроль над ней. Прежде чем жать Submit, всегда спрашивайте себя: доверяете ли вы организации, получающей данные? Если нет, лучше не посылать.

HTTPS – прекрасный способ сделать вашу работу в Интернете конфиденциальной. Однако обеспечить, чтобы использовался именно он, не всегда легко из-за его привязки к HTTP. Например, если использовать <https://www.google.com> для поиска “wikipedia”, вас направят на HTTP-энциклопедию, а не на HTTPS.

Некоммерческая организация Electronic Frontiers Foundation (EFF), созданная для охраны цифровых прав, разработала расширение для *Firefox*, которое вынуждает браузер использовать именно HTTPS, если таковой доступен. Версия для *Chrome* пока в состоянии бета. Возьмите расширение с <https://www.eff.org/https-everywhere>, чтобы ваша работа в Интернете была избавлена от перехватчиков.

Как и все виды шифрования, SSL имеет слабость: она в том, что ключи хранятся в сертификатах. И подобно тому, как, зная ваш пароль, взломщик может войти в вашу учетную запись, он же



► Коварный сайт автора (нижняя строка) использует тот же значок, что и подлинный безопасный сайт (верхняя строка), усыпляя бдительность пользователей.

`http.request.full_uri contains "duckduckgo.com?q"`

Если сейчас вы запустите поиск на <http://duckduckgo.com>, он появится в списке, а предмет поиска будет стоять в столбце Info. Похожую технику можно использовать для любой популярной поисковой системы. Вас могут не волновать те, кто может прочесть текст запроса, однако именно такая техника применяется, чтобы узнать имена пользователей и пароли, передаваемые открытым текстом. А так, между прочим, посылает пароли большинство форумов (поскольку они не представляют большой угрозы безопасности, а сертификат безопасности стоит дорого). Форумы на www.linuxformat.com устроены именно таким образом.

Чтобы разведать пароли с LinuxFormat.com, запустите *Wireshark* и начните захват пакетов со следующим фильтром:

`http.request.uri contains "login.php"`

Когда вы зайдете на www.linuxformat.com/forums/index.php (вам придется завести учетную запись, если у вас ее пока нет), этот фильтр захватит пакет. Текстовая строка будет содержать:

`Username=XXX&password=YYYY&login=Log+in`

Какие еще компьютеры имеют доступ к этой информации? В зависимости от настроек вашей сети, возможно, это все компьютеры в LAN или

беспроводной сети. А также – все компьютеры, находящиеся между вами и сервером, с которым вы связываетесь. Чтобы узнать, кто они, скопните `tracert` – отобразится маршрут ваших пакетов. Например:

`tracert www.google.com`

Если ваш компьютер прикрыт брандмауэром, вывод этой команды может состоять из одних звездочек. В таком случае можно воспользоваться сетевым `tracert` – например, из тех, что пе-

«Главная часть головоломки web-безопасности — Secure Sockets Layer.»

речислены на www.tracert.org. Этот список немного устарел, и `tracert` размещен не на всех серверах, однако вы можете отыскать такой, что работает в вашей области.

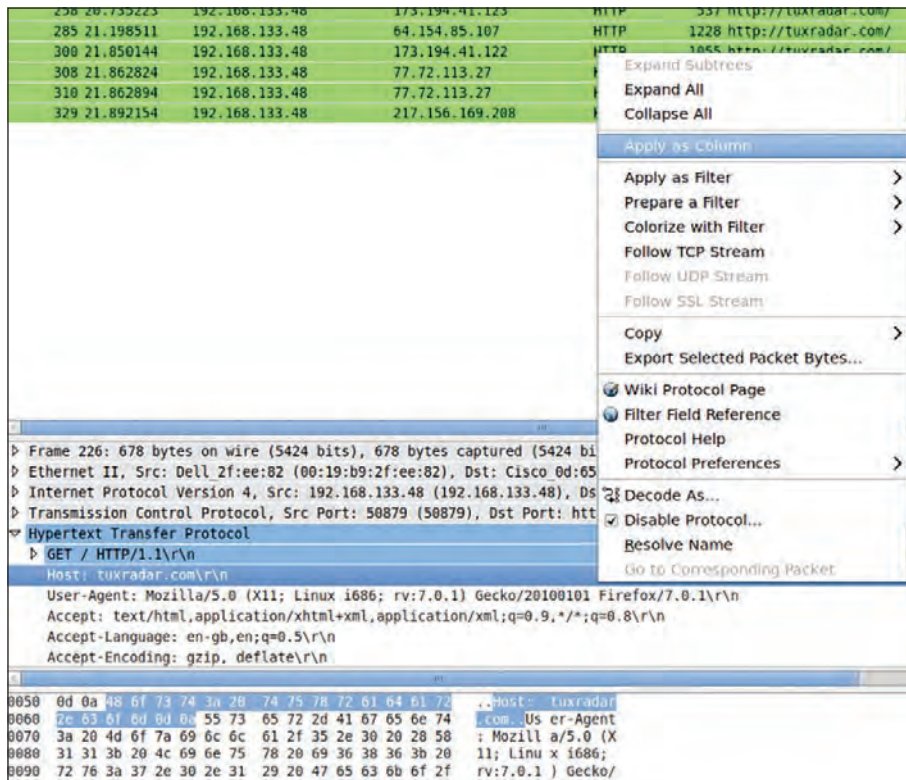
Вы знаете, чьи это компьютеры? Или тех, кто имеет к ним удаленный доступ? Оно вам надо, чтобы эти люди видели все, что вы делаете онлайн?

Если вы пользуетесь сервисами с незащищенными паролями (а отказываться от них нет причины, если вы вполне осознаете последствия), важно не применять для них тот же пароль, что и для безопасного, защищенного сервиса.

Главная часть всей головоломки web-безопасности – Secure Sockets Layer (SSL). Этот протокол с туманным названием – способ создания зашифрованного канала между программой, запущенной на вашем компьютере, и программой, работающей на другом компьютере. На любой небезопасный сетевой протокол найдется безопасный, который выполняет ту же задачу, но через канал SSL. Подробности приведены в таблице слева. Всякий раз, когда вы обращаетесь к протоколу из левого столбца, перехватчик может прочитать то, что

Опасно/Сохранно

Небезопасный	Использует SSL
FTP	SFTP
Telnet	SSH
RCP	SCP
HTTP	HTTPS



➤ Рис. 1. *Wireshark* позволяет добавить к описанию пакетов дополнительные столбцы.

легко перехватит или подделает данные, зашифрованные при помощи SSL, если «убедит» ваш компьютер использовать ложные сертификаты. Штука в том, что сертификаты, в отличие от паролей, хранятся в компьютере, а не в вашей голове. Поместив в вашу систему свои файлы, взломщик сможет взломать SSL-шифрование. Особому риску вы подвергаетесь, пользуясь компьютером, где не вы сами устанавливали операционную систему – например, на рабочем месте или в интернет-кафе.

Вы должны уметь просматривать текущие сертификаты в настройках безопасности вашего браузера, хотя не так просто распознать то, чего там быть не должно. Здесь вам помогут live-дистрибутивы: вы можете прихватить с собой надежную операционную систему и пользоваться ею всякий раз, когда вынуждены садиться за сомнительный компьютер.



Служка

SSL обеспечит защиту ваших данных от перехватчиков, но что делать, если компания, с которой вы общаетесь, шпионит за вами?

Google, Facebook, Twitter и другие сетевые гиганты построили свои бизнес-модели на предоставлении пользователям бесплатного сервиса в обмен на информацию о них, которая применяется, чтобы нацелить на пользователей адресную рекламу. Twitter зашел еще дальше и продал твиты пользователей маркетологам. Некоторые считают это честной сделкой, однако сторонники конфиденциальности информации все больше тревожат увеличение объема данных, которыми

завладевают эти компании, что выходит далеко за пределы предоставляемого им добровольно. И Google, и Facebook установили взаимосвязь с буквально миллионами других сайтов, чтобы помочь им отслеживать ваши перемещения по Ин-

«Twitter зашел дальше и продал твиты пользователей маркетологам.»

тернету через cookie. Несмотря на вкусное название [cookie – *англ.* печенюшка], на самом деле это информация, хранимая на вашем компьютере, чтобы сайты могли идентифицировать вас, когда вы заходите на них через свой браузер. Определить, сколько именно компаний следит за нами,

можно при помощи *Wireshark*, отслеживая сетевые соединения и подсчитав, сколько вернулось cookie.

Запустите *Wireshark*, захватите главный сетевой интерфейс. В поле фильтра введите `http.cookie`

Показаны будут только пакеты, связанные с cookie, отсылаемыми web-серверам. Чтобы узнать побольше о передаваемой информации, перейдите на среднюю панель и нажмите на кнопку рядом с Hypertext Transfer Protocol. В протоколе есть два раздела, которые позволяют web-компаниям следить за нами: `host` и `referer`. Щелкните правой кнопкой по каждому из них и выберите `Apply As Column` (см. рис. 1). Эти поля добавятся в главный просмотр. Каждый из этих двух доменов позволяет хосту (организации, получающей cookie) отслеживать вашу активность на реферере. Вдобавок хост использует уникальный ID, чтобы отслеживать вашу активность между сеансами.

Для отслеживания наших действий Google использует свои рекламные службы, а Facebook применяет кнопку Like. Узнать, что именно эти компании делают с нашими данными, невозможно: мы видим лишь то, что они получают.

К счастью, многие браузеры позволяют управлять cookie. В зависимости от ваших личных вкусов, вы можете ограничить cookie только определенными сайтами (там, где они удобны для запоминания паролей) или вообще заблокировать их. Если вы пользуетесь *Firefox*, зайдите в `Edit > Preferences > Privacy` и поменяйте `Firefox Will not Use Custom Settings for History`. Если снять галочку с `Accept Cookies From Sites`, никаких cookie *Firefox* хранить не будет. Чтобы проделать это в *Chromium*, зайдите в `Preferences` (гаечный ключ рядом с строкой адреса) > `Under the Bonnet` и **измените Cook-**

ies на `Block Sites From Setting Any Data`. В *Konqueror* это делается через `Settings > Configure Konqueror > Cookies`, отключением `Enable Cookies`. Для пользователей легковесного KDE это производится в *Rekonq*, в `Settings` (гаечный ключ рядом с строкой адреса) > `Network > Cookies` и отключением `Enable`

Когда Tor не работает

Если вы уже прочли раздел про Tor и у вас сложилось мнение, что это идеальный способ обновить вашу коллекцию фильмов/музыки на халяву без риска быть пойманным, подумайте еще раз. Есть две главных причины так не поступать: во-первых, это перегрузит сеть и замедлит ее работу для законных пользователей; во-вторых, может и не сработать, так как многие популярные протоколы обмена файлами прихватывают ваш IP-адрес. Анонимизация в Tor препятствует внесению вас в качестве отправителя в заголовок TCP-пакета, однако эта информация в данных есть.

Это вроде как с электронной почтой. Если у вас анонимный адрес электронной почты, заголовок не раскроет, кто вы такой, но если вы впишете адрес в тело письма, то получатель будет знать, где искать.

Простой способ обеспечить свою анонимность онлайн – использовать Интернет только через официальный браузерный пакет Tor, а не любое другое приложение. Используйте web-почту вместо электронной и сторонитесь документов, способных содержать интернет-ссылки (например, файлов DOC и PDF).

Использование открытого ПО

Возможно, самый большой риск для безопасности – позволить работать на вашем компьютере коду, который нельзя проверить. Так, в результате исследований в 2011 году некоторые пользователи iOS обнаружили, что их устройства втихомолку отслеживают их перемещения. Власти также использовали проприетарные системы, чтобы запрятать функции слежки.

Также в 2011, журнал «Экономист» написал, что некая американская компания продала правительству Саадма Хусейна фотокопии, которые,

подобно многим другим таким устройствам, были укомплектованы проприетарной системой. В этом пакете был упрятан способ передавать GPS-координаты армии США. Когда Джордж Буш запустил операцию «Нерушимая свобода», эти устройства-шпионы пришли в действие и направляли ракеты прямо в правительственные здания Багдада. Вся история здесь: www.economist.com/node/18527456.

Ричард Столлмен [Richard Stallman] ограничивает свое оборудование только тем, что использует всецело свободное ПО, от BIOS до всех драйверов

устройств. Из-за этого он загружается при помощи PMON и использует Lemote Yeelong и разновидность Linux под названием gNewSense (наши извинения мистеру Столлмену: он придет в ярость, прочитав это, но для краткости мы опускаем GNU!).

Хотя использование открытого ПО сможет гарантировать, что в коде нет гремлинов, большинству пользователей фактически невозможно удостовериться, что в оборудовании нет злонамеренных деталей. Единственный способ защитить себя – брать устройства из надежных источников.



➤ Можно удалить вечный cookie LSO, но при следующем посещении сайта он возродится опять.

Cookies. Наряду с полной блокировкой cookie, Firefox и Chromium предоставляют вариант блокировки cookie третьих лиц (в Konqueror и Rekonq, этому соответствует Only Accept Cookies From Originating Server). Это значит, что они блокируют cookie с доменов, отличных от текущего сайта. Если поступить так, сайты смогут хранить данные о вас – например, ваши предпочтения – и отслеживать ваши перемещения по сайту, но другие сайты не увидят ваши перемещения, когда вы покинете доменную зону. Это предотвратит отслеживание компаниями ваших перемещений в сети.

Если выполнить эти установки и снова запустить отслеживание cookie в Wireshark, вы увидите, что сейчас referer и host – всегда один и тот же домен. Для многих пользователей это станет золотой серединой: пусть cookie выполнят свою изначальную задачу – позволяют сай-



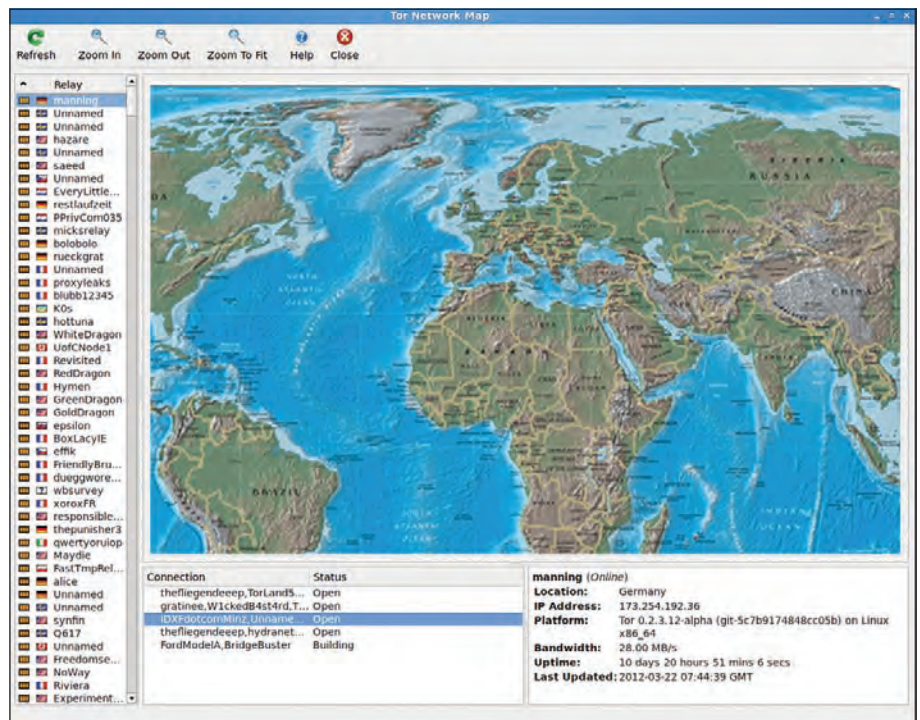
➤ Hostip.info смог определить наш IP-адрес в миле от штаб-квартиры LXF.

там распознавать постоянных пользователей; зато отслеживание разными организациями перемещений пользователей заблокировано. Cookie – не единственный способ слежения сайтов за вами. Даже если cookie в вашем браузере выключены, сайты все равно могут хранить отслеженную информацию на вашем компьютере, используя Locally Shared Objects (LSO). Они работают в точности как cookie, только доступ к ним идет через Flash, а не через ваш браузер. Чтобы распознать и контролировать сайты, которые это используют, зайдите на Website Storage Settings Panel в Macromedia по адресу www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager07.html.

«Web-мастера, нацеленные на слежку за вами, могут создать cookie-зомби.»

Web-мастера, нацеленные на слежку за вами, могут использовать сочетание методов, чтобы создать cookie-зомби. Они хранят одну и ту же информацию в нескольких местах, и если вы удаляете одно, информация воссоздается из других. Например, если вы удалите все cookie браузера, сайт воссоздаст cookie из LSO, и наоборот. Пока остается один из них, все остальные смогут восстановиться. Сами Камкар [Samy Kamkar] дошел до крайности в своем samy.pl/evercookie, применив 12 различных методов воссоздания данных!

Мы думали, запуск расширения NoScript для Firefox предотвратит работу cookie такого типа, но оно



➤ Программа Vidalia, поставляемая с браузерным комплектом Tor, покажет, как анонимное соединение путешествует по свету.

Вредоносное ПО [malware]

Вирусы и трояны – сфера не только киберпреступников и технарей, доказывающих концепцию. Власти также могут использовать их для слежки за населением, и они действительно так поступают. WikiLeaks опубликовал документы 2008 года, где говорилось, что полиция и прокуратура Баварии поделили оплату системы перехвата зашифрованных звонков Skype. Впоследствии правительство признало использование программ для перехвата звонков. Примененный ими троян захватывал данные с компьютера еще до шифрования.

Более свежий пример – использование сирийским правительством вредоносного ПО для доступа к компьютерам диссидентов. Эта программа умела записывать все нажатия клавиш пользователей и включать их web-камеры, что позволило режиму связать людей с анонимными учетными записями в Интернете, а также разоблачать участников онлайн-форумов.

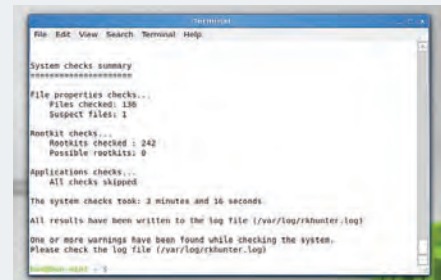
Чтобы избежать этой угрозы, следуйте обычным предосторожностям против вредоносного ПО. Хотя пока неизвестны вирусы, способные нанести вред современному компьютеру с Linux (те немногие,

что существовали, использовали ныне закрытые дыры в безопасности), для троянцев это уже не так. Как было показано в LXF154, для Linux их создать легко.

Проблема с троянами в Linux попала в новостные ленты в марте 2012, когда неизвестные выпустили Anonymous-OS. Вскоре после релиза, @YourAnonNews объявил в Твиттере, что она «облепена троянами». Однако программы аналитики безопасности заняли на этот счет странную позицию: большинство лишь бормотало о неразумности брать ПО из непроверенных источников.

Это подчеркнуло проблему, что идентификация вредоносного ПО в Linux не такая зрелая, как в Windows. Утилиты вроде *rkhunter* распознают только самые основные трояны; они не заметили троян, созданный нами в LXF154. Одна проверка подтвердила, что пакет – с официального сервера и не является подделкой, однако немало других мест, где можно спрятать вредоносное ПО. Запуск Tor усугубляет проблему. Тут вам уже не отследить трафик, исходящий из вашего компьютера – он зашифрован и безымянен. Мораль этой истории – важно,

чтобы на ваш компьютер с Linux не попало вредоносное ПО, так как без качественной антивирусной программы его очень сложно отловить, не говоря уже об избавлении от него. Самые важные способы защиты вашей системы – установка только тех программ, что подписаны надежным источником (таким как репозиторий вашего дистрибутива) и использование полного шифрования диска, чтобы на него не проник вредоносный агент.



➤ Команда 'sudo rkhunter --check' проверит систему на наличие любого известного malware.

также делает негодным метод проверки! Мы обнаружили, что ни режим Private в Firefox, ни режим Incognito в Chromium не могут этого предотвратить. Если вы хотите гарантировать, чтобы ваша работа в Интернете не отслеживалась между сеансами, лучшим решением будет использовать невозобновляемую [non-persistent] систему – то есть не сохраняющую никакой информации по завершении сеанса. Так вас можно будет отследить во время сеанса, но не между сеансами. Для пользователей Linux самым очевидным

решением является live DVD. Это не обязательно должен быть физический носитель – достаточно ISO, запущенного на виртуальной машине. При этом все данные, по которым сайты смогут отслеживать вас, будут сбрасываться при каждом перезапуске виртуальной машины. Также можно запускать одновременно несколько виртуальных машин, чтобы никто не мог связать два сеанса. Если это произойдет, для таких целей очень удобна live-версия Boot To Gecko, однако она все еще в разработке.

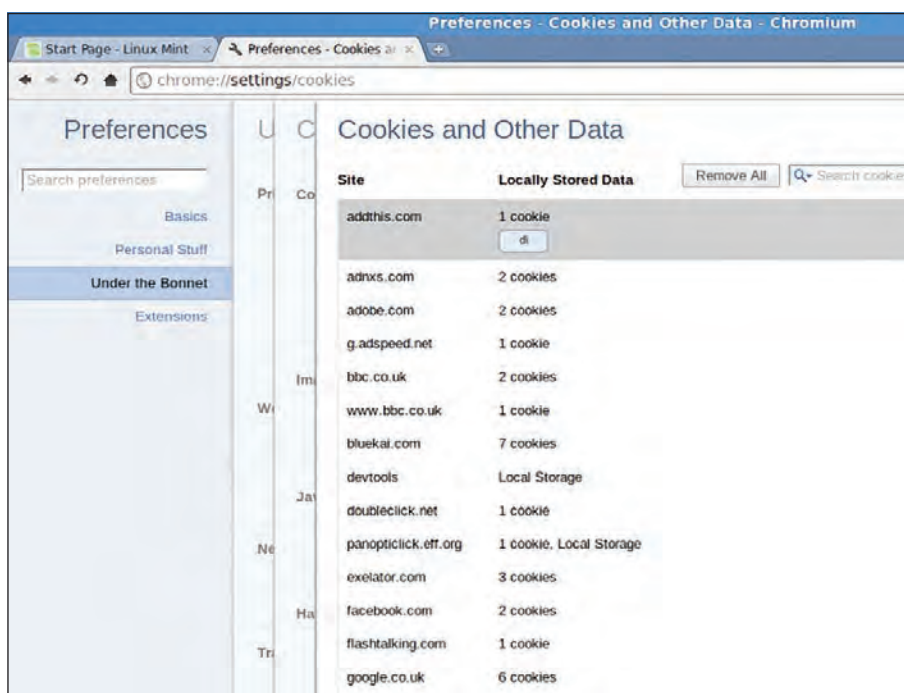


Цифровые отпечатки

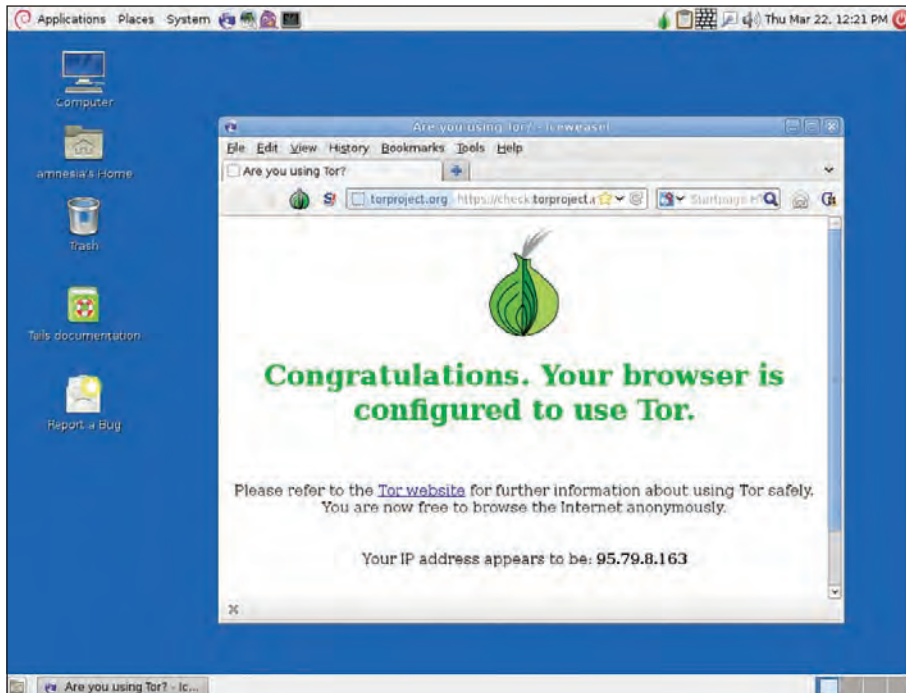
Это еще одна методика, немного похитрее, по которой сайты смогут идентифицировать вас. Она собирает во едино всю информацию о возможностях вашего браузера и системы в цифровой отпечаток. Из-за объема информации о вас, которой делится браузер при запросах, этот отпечаток часто может быть использован, чтобы однозначно идентифицировать вас на сайте.

И снова, в этой области проявляет активность EFF: они создали сайт, позволяющий оценить ваш отпечаток. Чтобы узнать, насколько вы уникальны, зайдите на panopticklick.eff.org. На момент написания статьи более двух миллионов человек обратились к этому сайту для проверки своих браузеров, и мы все еще видим, что большинство наших компьютеров однозначно идентифицируются. А значит, любой сайт может следить за нами, даже не прибегая к cookie, LSO и другим хранимым данным. На данный момент это только теоретически возможная уязвимость, и случаи ее бесконтрольного использования еще не встречались.

Если вас беспокоит такой способ слежки, лучший способ предотвратить ее – отключить запуск скриптов. Это существенно снизит количество информации, по которой сайт формирует отпечаток. Расширение NoScript для Firefox предоставляет простой способ управлять запуском скриптов с сайта. Однако это сильно ограничит функциональность многих интерактивных сайтов. Web-страницы состоят из большого числа различных элементов, которые браузер переупорядочивает, формируя единый документ. Эти элементы происходят из разных мест, организаций и серверов. Все они могут содержать некоторую степень слежки по технологии web-маяк (также известной



➤ Автор обнаружил 40 cookie в Chromium, а ведь это не основной его браузер.



» Tails (на DVD этого месяца) поможет пресечь слежку за вашими действиями онлайн.

как pixel tags). Они используют изображения для генерации HTTP-запросов, которые записывают ваши действия на разных серверах на тот, где располагается сайт. Потенциально это может использоваться для слежки, в сочетании с отпечатком браузера, но применяется более широко, не ограничиваясь web-страницами: это можно включить в любой документ HTML. В основном это используется спамерами для определения активных адресов электронной почты. Если вы открываете электронное письмо, содержащее подобное изображение,

спамер автоматически определит, что почта проверяется и на нее можно отправить спам-рассылку.

К счастью, в большинстве почтовых клиентов и в web-почте загрузка изображений по умолчанию отключена.



Обнаружение

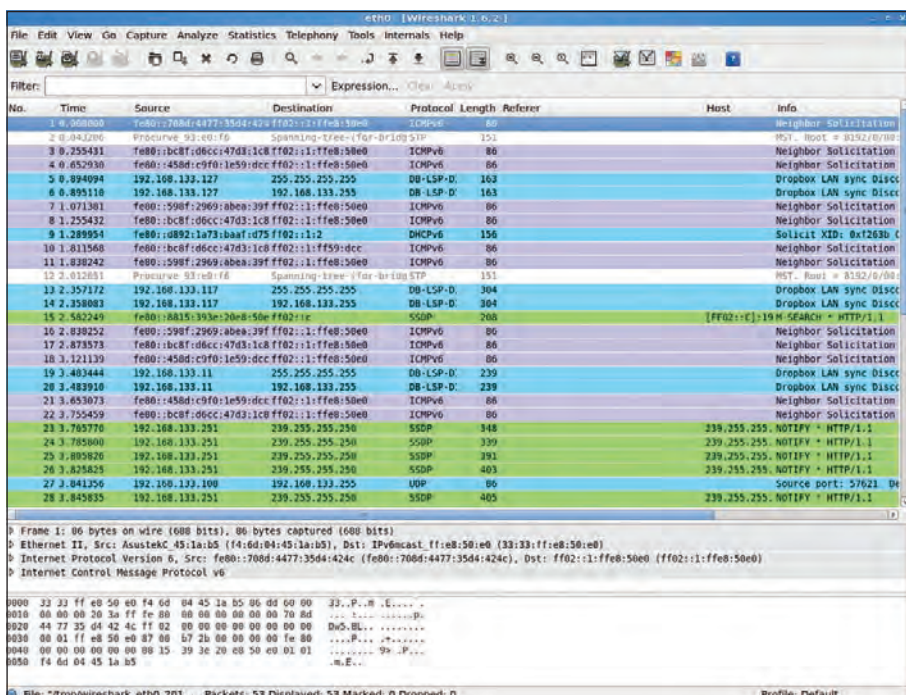
При подключении к Интернету ваш провайдер присваивает вам адрес IP (Internet Protocol). Он показывает

web-серверам и другим компьютерам, с которыми вы соединяетесь, куда отсылать информацию. Любой компьютер, с которым вы общаетесь онлайн, сможет сказать, какой у вас IP-адрес, и определить по нему какую-либо информацию, в основном о вашем провайдере и примерном местоположении. Зайдите на www.hostip.info и узнайте, что именно вы рассказываете миру. Поскольку IP-адреса периодически меняются, web-серверы не смогут подступиться к вам ближе этого места. Однако правительственные службы могут заставить вашего провайдера сообщить, какой именно клиент приписан к какому IP-адресу в любой момент. Короче говоря, они могут связать онлайн-действие с определенным компьютером.

Например, в апреле 2004 года китайский журналист Ши Тао [Shi Tao] написал электронное письмо в Asia Democracy Foundation с подробностями о попытках китайского правительства заглушить сообщения о 15-летней годовщине событий на площади Тяньаньмынь в web-почте Yahoo. Его правительство узнало IP-адрес, которым он пользовался в Yahoo, а поскольку в Китае провайдером является государство, они смогли в точности узнать, откуда ушло письмо. В ноябре журналист был арестован, а в марте 2005 года засажен в тюрьму на десять лет.

Чтобы защитить себя на этом уровне безопасности, позаботьтесь, чтобы между вами (и вашим IP) и сервером, с которым вы связываетесь, отсутствовала прямая связь. Простого шифрования вашего соединения будет недостаточно: сервер все равно узнает, кто источник, это спасает только от перехватчиков. Необходимой приватности можно достичь, передавая свои данные через эстафету шифрованных узлов [relays]. Такая технология называется луковичной, или многослойной, маршрутизацией [onion routing], и она реализована в проекте Tor.

- » Шаг один: Свяжитесь с сервером каталога Tor, который выдаст вам три случайных узла.
- » Шаг два: Зашифруйте ваши данные с помощью ключей от каждого из узлов.
- » Шаг три: Отшлите зашифрованный пакет первому узлу. Этот сервер знает ваш IP-адрес,



» Wireshark охватывает широкий спектр информации. К счастью, здесь также предусмотрены фильтры, чтобы вы смогли в ней разобраться.

Проект I2P

Проект I2P нацелен на создание анонимной и безопасной сети. В отличие от Tor или HTTPS, он не связан с Интернетом, а представляет отдельную одноранговую сеть. Программа пока еще на бета-стадии, и ее не следует использовать в ситуациях, когда анонимность является условием выживания. Однако проект растет и развивается, и, надеемся, в будущем станет еще одним оружием в войне против цифровых вторжений.

но не знает, что вы делаете, так как данные зашифрованы ключами от других узлов. Все, что они могут узнать – это расположение второго узла.

» **Шаг четыре:** Первый узел отправляет зашифрованный пакет второму узлу, который может дешифровать только местоположение третьего узла. Этот компьютер знает о двух других узлах, но не знает ни ваш IP, ни с кем вы хотите связаться.

» **Шаг пять:** Второй узел посылает зашифрованный пакет третьему. Этот компьютер может дешифровать ваше сообщение и послать его в сеть Тог получателю. Третий узел может увидеть получателя ваших данных (а если вы используете нешифрующий протокол, то и сами данные), и положение второго узла, но не знает, кто вы такой.

» **Шаг шесть:** Адресат получает ваш запрос, как если бы он пришел от третьего узла. Они не знают ни кто вы такой, ни даже того, что кто-то скрыт за третьим узлом. Они отвечают третьему узлу.

» **Шаг семь:** Третий узел проводит всю информацию обратно по сети Тог тем же способом, как она была послана. Никто в цепочке не знает одновременно личности отправителя и получателя. Однако Тог – это система анонимизации, а не шифрования. Пока данные проходят через узел, они шифруются, однако, покинув сеть, они становятся ничуть не защищеннее, чем любая другая информация в Интернете. Чтобы данные были конфиденциальными, применяйте те же

«Проект Тог поместил все необходимые инструменты в один пакет.»

предосторожности, которые рекомендуются без Тог, то есть используйте шифрующие протоколы, показанные в правом столбце таблицы. Выглядит мутным? К счастью, проект Тог поместил все необходимые для этой цели инструменты в один пакет с безопасной версией *Firefox*. Он есть и на диске, и на сайте www.torproject.org – просто распакуйте файл и скомануйте **start-tor-browser**. Вы подключитесь к Сети, и откроется защищенный браузер. Если вы находитесь в бегах (в любом смысле этого слова), можете безопасно зайти в Интернет, воспользовавшись *Orbot* для Android или *Covert Browser* для iOS.

Browser Characteristic	bits of identifying information	one in x browsers have this value	value
User Agent	9.89	947.24	Mozilla/5.0 (X11; Linux i686; rv:7.0.1) Gecko/20101010 Firefox/7.0.1
HTTP_ACCEPT Headers	8.24	302.15	text/html,*/* ISO-8859-1,utf-8;q=0.7,*;q=0.7,gzip, deflate,en-gb,en;q=0.5
Browser Plugin Details	16.66	419625.2	Plugin 0: Cinnamon Integration; This plugin provides integration with Cinnamon for live extension enabling and disabling. It can be used only by extensions.gnome.org; ibcinnamon-browser-plugin.so; (Cinnamon Integration Dummy Content-Type: application/x-cinnamon-integration: 1); Plugin 1: GNOME Shell Integration; This plugin provides integration with GNOME Shell for live extension enabling and disabling. It can be used only by extensions.gnome.org; libgnome-shell-browser-plugin.so; (GNOME Shell Integration Dummy Content-Type: application/x-gnome-shell-integration: 1); Plugin 2: Java(TM) Plug-in 1.6.0_26; The next generation -> a href="http://java.sun.com/">http://java.sun.com/ -> Java plug-in for Mozilla browsers; libnp2.so; (Java™ Plug-in; application/x-java-vm; (Java™ Plug-in Applet; application/x-java-applet; (Java™ Plug-in; application/x-java-applet-version=1.1; (Java™ Plug-in; application/x-java-applet-version=1.1.1; (Java™ Plug-in; application/x-java-applet-version=1.2; (Java™ Plug-in; application/x-java-applet-version=1.2.1; (Java™ Plug-in; application/x-java-applet-version=1.2.2; (Java™ Plug-in; application/x-java-applet-version=1.3; (Java™ Plug-in; application/x-java-applet-version=1.3.1; (Java™ Plug-in; application/x-java-applet-version=1.4; (Java™ Plug-in; application/x-java-applet-version=1.4.1; (Java™ Plug-in; application/x-java-applet-version=1.4.2; (Java™ Plug-in; application/x-java-applet-version=1.5; (Java™ Plug-in; application/x-java-applet-version=1.6; (Java™ Plug-in; application/x-java-applet-version=1.6.0_26; (Java™ Plug-in JavaBeans; application/x-java-bean; (Java™ Plug-in; application/x-java-bean-version=1.1; (Java™ Plug-in; application/x-java-bean-version=1.1.1; (Java™ Plug-in; application/x-java-bean-version=1.2; (Java™ Plug-in; application/x-java-bean-version=1.2.1; (Java™ Plug-in; application/x-java-bean-version=1.2.2; (Java™ Plug-in; application/x-java-bean-version=1.3; (Java™ Plug-in; application/x-java-bean-version=1.3.1; (Java™ Plug-in; application/x-java-bean-version=1.4; (Java™ Plug-in; application/x-java-bean-version=1.4.1; (Java™ Plug-in; application/x-java-bean-version=1.4.2; (Java™ Plug-in; application/x-java-bean-version=1.5; (Java™ Plug-in; application/x-java-bean-version=1.6; (Java™ Plug-in; application/x-java-bean-version=1.6.0_26; Plugin 3: Shockwave Flash; Shockwave Flash 11.1 r102; libflashplayer.so; (Shockwave Flash; application/x-shockwave-flash; swf) (FutureSplash Player; application/futuresplash; spl) Plugin 4: Skype Buttons for Kopete; Mime Type x-skyype for Skype Buttons; skypebuttons.so; (Skype Buttons; application/x-skyype; 1;
Time Zone	4.53	23.17	0
Screen Size and Color Depth	4.21	16.51	1280x1024x24
System Fonts	16.42	349667.67	Junkyard, Century Schoolbook L, OpenSymbol, progenis, Il wasn't me, Droid Sans Mono, Domestic Manners, DejaVu Sans Mono, Swill, Marked Foot, URW Palladio L, Unifont, URW Gothic L, Dingbats, URW Chancery L, Ubuntu, FreeSerif, Ubuntu Condensed, DejaVu Sans, El Abogado Loco, WenQuanYi Micro Hei Mono, Winks, DejaVu Serif, Wargames, Symbol, Bitstream Charter, DejaVu Sans, Courier 10 Pitch, Liberation Sans Narrow, Liberation Mono, Nimbus Sans L, Baskerville, WenQuanYi Micro Hei, Droid Serif, Standard Symbols L, Nimbus Mono L, Liberation Serif, Nimbus Roman No9 L, Ubuntu, Liberation Sans, FreeSans, Penguin Attack, flatline, Duzismo Roman, DejaVu Serif, Duzismo, FreeMono, Ubuntu Mono, URW Bookman L, Droid Sans (via Freetype)
Are Cookies Enabled?	0.39	1.31	Yes

» Это информация для любопытного web-мастера, чтобы отследить автора по отпечатку браузера.

Существует потенциальная статистическая уязвимость Сети. Например, если организация видит все входящие в сеть данные и все данные, исходящие из нее, время и количество пакетов может выдать их отправителя. Однако из-за всемирного характера Интернета это потребует скоординированного и систематического мониторинга во всех странах мира.

Можно вообразить, что использование учетной записи, не привязанной к физическому местоположению, например, с мобильного или спутникового телефона, улучшит ситуацию. Наоборот! Сигналы мобильного телефона подпадают под триангуляцию, а многие спутниковые телефоны включают в связь с провайдером передачу GPS-координат местоположения телефона.

Польская фирма TS2 торгует продуктом, способным указать пользователя спутникового телефона: www.ts2.pl/en/News/1/151. Возможно, сходная технология использовалась режимом в Сирии, чтобы засечь и уничтожить журналистов

в Хомсе в начале этого года. Прочитайте вкладку «Когда Тог не работает», там рассказано подробнее, почему использование Тог не защитило бы их от надзора.

Некоторые режимы, прежде всего в КНР, приняли явные шаги, чтобы лишить своих граждан доступа к Тог. Самый простой способ сделать это – скачать список узлов Тог и перерезать связь с этими компьютерами. Чтобы пользователи смогли обойти это препятствие, Тог ввел несколько сетевых мостов. Это маршруты в сети, которые не публикуются. Сейчас началась игра в кошки-мышки между проектом Тог и организациями, пытающимися блокировать доступ к сервису анонимизации.

Подобно многим проектам, развиваемым сообществом, Тог нуждается в добровольцах. Однако самые востребованные здесь не программисты, что необычно для программного проекта. Обслуживание узла или моста поможет людям сохранять свою анонимность. Также требуются переводчики и юристы. Чтобы увидеть, как можно помочь людям отстаивать свою приватность и свободу слова, посетите сайт проекта – www.torproject.com/getinvolved/volunteer.

Офисная приватность

Даже если действия совершаются не в Интернете, это не значит, что их нельзя отследить. Есть как минимум теоретическая возможность отследить по серийным номерам DVD, CD, некоторые USB-брелки и другие устройства хранения. По номеру государственные учреждения могут отследить их до распространителя и, в зависимости от способа

оплаты, и до заказчика. Если вы намерены обнаружить информацию через сервис вроде Wikileaks с помощью офлайн-методов (например, обычная почта), позаботьтесь, чтобы такие детали не вывели на вас – нет, мы не считаем, что Wikileaks на вас донесет, но информация может быть перехвачена. Чтобы избежать риска, покупайте устройства (в том

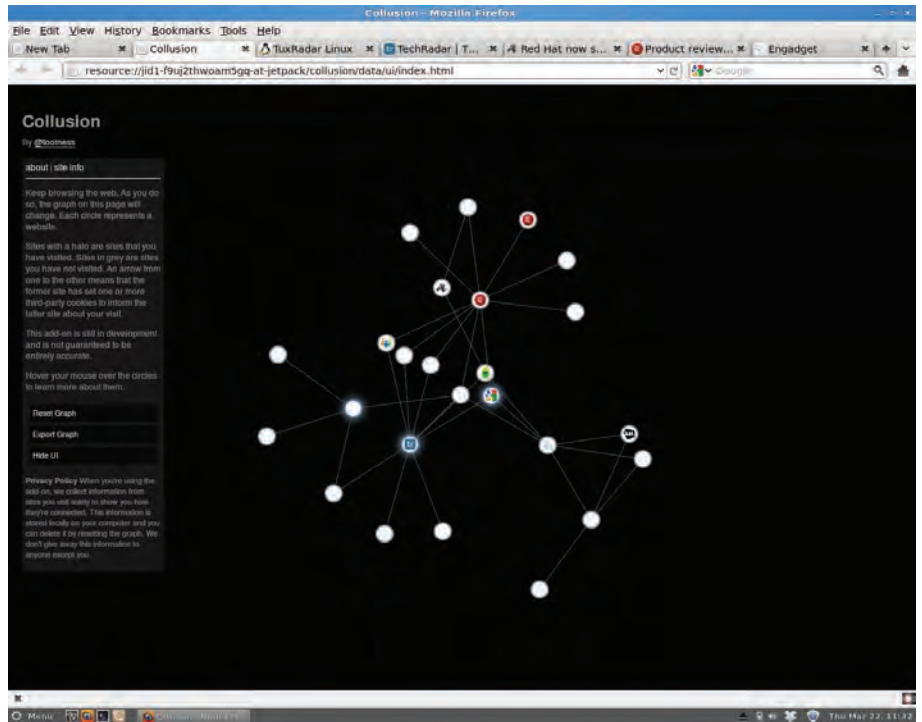
числе для прожига CD/DVD) за наличные и не из дома. И не отправляйте посылку через пункт, который можно связать с вами, то есть через ближайший к дому почтовый ящик или почтовое отделение со скрытой камерой.



GPG

Электронная почта стала незаменимым средством связи. Однако она зародилась в те дни, когда еще не было электронного шпионажа и вопросы неприкосновенности личной жизни не стояли так остро. Содержимое вашей электронной почты посылается незашифрованным и может быть прочитано всяким, у кого есть доступ к сети. Использование web-почты по HTTPS не намного снизит количество людей, которые могут вскрыть письмо, так как ваша почта, прибыв на сервер почтового провайдера, сначала дешифруется и только потом отправляется получателю.

Желая обезопасить свою почту, шифруйте ее самостоятельно до того, как она покинет ваш компьютер. Простейший способ проделать это в Linux – GPG (Gnu Privacy Guard), который имеет модули расширения для многих популярных почтовых программ. Они предлагают простую в использовании методику скрытия содержимого вашей электронной почты от всех, кроме адресата. Как и при любом шифровании, безопасность зависит от качества используемых ключей (и их секретности).



➤ Расширение *Collusion Firefox* рисует граф рассылки информации cookie по сайтам. После установки расширения нажмите на белый кружок внизу справа для его запуска.



Шифрование диска

Если вы озабочены приватностью, то возможно, шифруете весь диск. Если нет, то вас это заинтересует.

Установить шифрование несложно: обычно это сводится к проставлению галочки при установке дистрибутива, а на современных системах сопутствующие затраты производительности пренебрежимо малы по сравнению с другими задачами. Учтите только, что частичное шифрование диска гораздо менее безопасно – в LXF154 мы продемонстрировали, как обмануть такую систему. Современные методы шифрования, используемые алгоритмы, подобные AES, невозможно взломать, не зная парольной фразы, и безопасность обеспечивается достаточно длинным ключом (как минимум, AES-128, а если вы под колпаком у ЦРУ, лучше AES-256). Существуют методы, по которым властные структуры могут добыть эту

фразу-пароль. К сожалению, простейшим (для властей) является попытка. Второй по легкости способ – попытаться угадать фразу, используя словарную атаку.

«Вопреки расхожему мнению, ОЗУ не сбрасывается при выключении.»

Но допустим, вы выбрали парольную фразу, которую невозможно угадать, и, заслышав стук в дверь, успели выпрыгнуть в окно и ударить. Уцелеют ли ваши секреты? Да не совсем.

При шифровании диска компьютер хранит ключи для дешифровки в оперативной памяти. Если преследователи вышибут дверь как раз в момент выключения компьютера, они смогут вставить в ваш компьютер

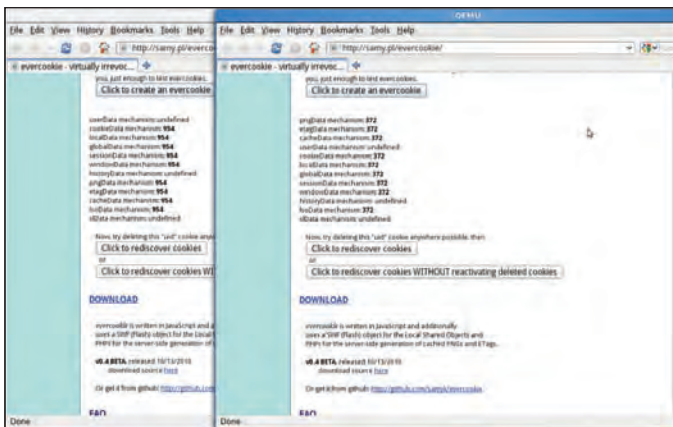
Если вы ограничились блокировкой или приостановкой вашего компьютера, ситуация еще ухудшится. Тогда у сотрудников спецслужб будет время охладить память перед перезагрузкой

компьютера (или переносом блока ОЗУ на другой компьютер для прочтения). При комнатной температуре память, как правило, становится нечитаемой через пару секунд. Если же ее охладить примерно до -50°C (что легко достигается применением дешевых аэрозолей), продолжительность жизни уцелевшей в памяти информации возрастет до нескольких минут.

Чтобы избежать атак данного типа, предотвратите доступ к физической памяти. Не оставляйте свой компьютер заблокированным или приостановленным. Если на нем есть важная информация, всегда выключайте его. И не допускайте загрузки с устройств, отличных от вашего жесткого диска. Это не позволит атакующим загрузить утилиты наподобие программы чтения USB от принстонской группы: за время, пока они будут обходить настройки безопасности BIOS, память станет нечитаемой.

Использование более длинного ключа шифрования также пойдет на пользу, поскольку в процессе чтения возникают небольшие ошибки. И чем длиннее ключ, тем больше вероятность таких ошибок.

Если личности в черном действительно сели вам на хвост, рассмотрите вариант работы на ноутбуке без аккумулятора. В этом случае вам достаточно будет выдернуть провод из розетки и задать стрелкача. LXF



➤ Даже самый хитрый *evercookie* не отследит вас между двумя виртуальными машинами.

Киаран О'Риордан



Пока мы весело резвимся со своими Linux-машинами, незримая армия несет свою службу, защищая нашу свободу. Киаран О'Риордан — один из ее бойцов: как человек, отдавший много сил фонду Free Software Foundation в работе над реформой патентной системы, а ныне активист движения End Software Patents — это один из самых умных и преданных борцов за свободу, к которой мы так привыкли. Мы нашли его на FOSDEM, в Брюсселе.

LXF: Большинство наших читателей в Великобритании и Евросоюзе наверняка считает, что патентов на ПО не существует. Но с ваших слов получается, что их все же можно принудительно применять, хотя прямого закона нет. Как так?

Киаран О'Риордан: В целом, право на регулирование патентов распределено между патентным бюро и законодательной и судебной властью, и потому, как всегда, между всеми тремя действует система сдержек и противовесов. Закон гласит, что компьютерные программы не патентуются.

Но в бюро патентов истолковали это очень странно. А именно, зацепившись за само понятие «компьютерные программы» — они, как таковые, патентованию не подлежат, но если вы решите запатентовать программу как запатентованное устройство, содержащее некую информацию... то есть назовете ее как-то по-другому, то это совсем другое дело. И хотя в законе такого не сказано прямо, прецеденты имеются уже давно, и вопрос патентоспособности перешел в суд, а позиция британского суда сводится к следующему: «Ну, этот закон трудно истолковать, и раз уж EPO [European Patent Office, Европейское патентное ведомство] делает это так,

мы не будем создавать разночтений с ними и согласимся».

Говорите, в Евросоюзе нет патентов на ПО? Мы тоже так думали, пока не встретились с End Software Patents.

Один судья даже как-то аргументировал необходимость взаимного соответствия решений, что само по себе смешно, поскольку патентное бюро всегда выносит свой вердикт до суда. Тогда, выходит, суду остается только согласиться с ними. И вот возникает такая странная ситуация: EPO не является ни частью Евросоюза, ни частью Великобритании, и никому не подконтрольно, но принимает решения за Верховный суд Соединенного королевства. Закон нужно менять, но это требует огромных вложений. Когда в 2002-м нам пришлось выработать EU Software Patents Directive [Директиву Евросоюза о патентах на ПО], люди ради этого бросали свою основную работу. Я сам сменил страну, и все это ради продвижения проекта, который был в итоге остановлен. Значит, для победы нужно еще больше. Проблема в том, что когда, кажется, уже все идет к нашей победе, ее противники обязательно постараются этого не допустить.

LXF: А это касается людей, которые достаточно знают о патентах на ПО и европейском законодательстве и говорят на словенском, или мальтийском, или на другом языке Евросоюза?

К. О'Р: Языковой барьер тут не главное. В основном, все говорят по-английски. Труднее всего объяснить самим разработчикам, как действует этот закон. Нужно помнить, что если в тексте есть слово «технология», то оно значит вовсе не то, что гласит словарь; а то, как это слово понял какой-то судья лет 10 тому назад. А теперь у нас есть сферы технологий, и есть TRIPS

(Trade Related aspects of Intellectual Property rights, Соглашение по торговым аспектам прав интеллектуальной собственности), согласно которому инновации в любой из них патентуются.

В результате, чтобы ПО патентованию не подлежало, нам надо доказать, что ПО не является технологией. Очевидно, что разработчикам такой подход покажется странным, но если вы обратитесь к законам об авторском праве, то, согласно Бернской конвенции об охране литературных и художественных произведений, ПО попадает под защиту как художественное произведение. То есть программа является авторской работой, и вовсе не обязательно технологией.

LXF: И такая ситуация по всему Евросоюзу?

К. О'Р: В настоящее время идет разговор о создании некоего общеевропейского патента. Европейский союз сейчас включает 23 языка. Дело в том, что при желании получить патент, действующий на всей его территории, вам придется регистрировать его в каждом национальном патентном бюро и на каждом языке, там принятом, так что переводить его нужно тоже 23 раза. Чтобы удешевить данную процедуру, хотя бы ввести этот единый патент. Иногда его предлагают публиковать только на английском, с последующим машинным переводом на другие языки. Иногда говорят — на английском, французском и немецком; тут обижаются Испания и Италия, а если включить их, то обидится Польша...

Чтобы этого избежать, Евросоюз решил публиковать все только на английском, или только на английском, французском и немецком, но чтобы эти патенты действовали во всем Евросоюзе; и переводить его только при наличии судебного иска. И, конечно, к тому времени будет уже слишком поздно, поскольку это значит, что кто-то уже закон нарушил; так





что будь хоть 20 000 патентов, но вы не говорите ни на одном из трех упомянутых языков, вы и знать о них не знаете.

LXF: Какой-то кафкианский абсурд...

К.О'Р: Да, но для нас это даже плюс, ведь это одна из причин, по которым проект до сих пор не принят. Его вновь и вновь откладывают. Для разработчиков свободного ПО это большой роли не играет, это скорее вопрос демократии для нас всех. Разработчиков касается прежде всего то, в чьем ведении окажутся эти централизованные патенты, в случае их принятия, потому как если их рассмотрением займутся государственные суды, то мы вернемся к прежней ситуации, когда Великобритания сказала «нет», Франция сказала «да», Германия – опять «нет», и т.д. Так что логически, следующим шагом должен стать централизованный суд. То есть специальный суд под контролем Европейского суда, или национальных ведомств, но тогда они, в свою очередь, будут рассмат-

ривать его как суд экспертов и будут вынуждены всегда с ним соглашаться, что мы и видели в Соединенном Королевстве. Значит, этот суд юристов-патентоведов как таковой вытеснит национальные. И кончится тем, что патентное бюро будет выдавать патенты, закон будет игнорироваться, а судам останется только все подтверждать.

LXF: Среди политиков есть такие, кто это понимает?

К.О'Р: Да, Европарламент – там сидят довольно умные люди. Это тот орган, который Free Software Foundation удалось убедить в 2003-м и 2005 году принять поправки о том, что программное обеспечение патентованию не подлежит. В 2003-м все наши поправки были при-

няты, и это была, по сути, полная победа. Если бы текст был принят как есть, сегодня Европа могла бы жить спокойно. Однако затем Совет Европы и Совет министров эти поправки отменили. Поэтому, когда в 2005-м Европарламент вновь взялся за их рассмотрение и был вновь готов их принять, они подумали: «Какой смысл, если Совет вновь их отклонит? К чему нам тогда эти поправки?» – и на этом все завершилось.

В этом тоже была своя победа для нас, ведь удалось избежать худшего варианта развития событий; но с другой стороны, именно этого хотели Microsoft, IBM и SAP – ведь они чували, что проигрывают, и просто перекрыли кислород всему проекту; и в итоге тоже радуются, что для них это не полное поражение. Нам же

видится, что эту проблему необходимо решать законодательным путем, но для этого нам требуется воля к победе и готовность доводить процесс до конца, чтобы правильный закон вступил в силу. **LXF**

О ПАТЕНТАХ НА ПО
«Чтобы ПО патентованию не подлежало, надо доказывать, что это не технология.»



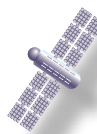
Прогулки — это хорошо. И велосело. А если вы хотите поделиться своими маршрутами, используя свои маршруты, используя GPS? **Иэн Бартон** покажет, как это делается.

КАРТА ВАШИХ ПРОГУЛОК

Будучи энтузиастом альпинизма и велоспорта, я использую GPS-навигатор для записи своих маршрутов, а цифровую камеру — чтобы запечатлеть виды. Желая поделиться маршрутами, мне надо показать свои перемещения на карте в web-браузере. Как и в Google Maps, мне надо было привязать мои фотографии к Point of Interest (POI, интересное место) на

карте и сделать так, чтобы фотографии всплывали, когда пользователь щелкает на POI. Весь процесс задействует огромную подборку инструментов с открытым кодом — от извлечения маршрутов из навигатора и снабжения фотографий геотэгами по данным GPS до отображения и маршрутов, и фото на карте при помощи библиотеки JavaScript, OpenLayers.

Загрузка данных



Первый шаг — извлечение данных с GPS-навигатора. У меня Garmin eTrex с последовательным портом. Однако многие более новые устройства снабжены портами USB. Если ваше устройство GPS имеет последовательный порт, вам, вероятно, понадобится стыковать его с USB через кабель. Найти такой кабель можно на eBay.

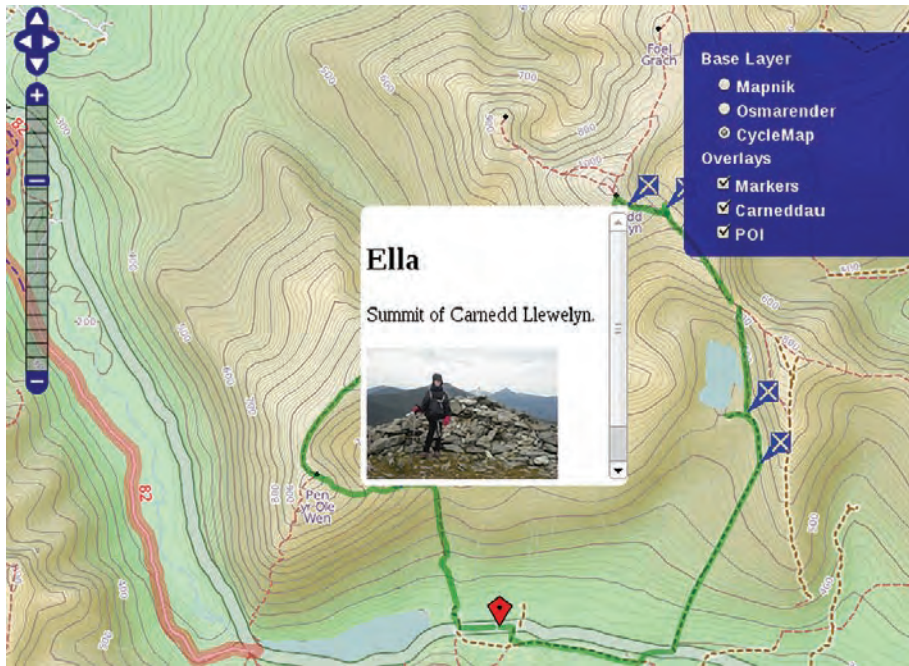
Для загрузки маршрутов и точек следования из GPS мы применим *GPSBabel*. Эта программа — настоящий швейцарский нож в мире GPS.

Она умеет конвертировать пункты маршрута и пути следования в самые разные форматы. Если вы используете конвертор из последовательного порта в USB, то при подключении его к вашему компьютеру и запуске *dmesg* вы должны получить нечто вроде

```
[ 5753.489759] usb 5-1: pl2303 converter now attached to ttyUSB0
[ 5753.489786] usbcore: registered new interface driver pl2303
[ 5753.489789] pl2303: Prolific PL2303 USB to serial adaptor driver
```

Используя GPS от Garmin, вы получите наилучший результат при загрузке необработанного журнала трассы [tracklog]. Сохраненные журналы теряют изрядную долю информации, в том числе отметки времени. Но если вы хотите сделать геотэги для своих фотографий, отметки времени вам понадобятся!

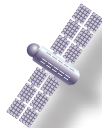
Лучше всего сохранять данные в формате GPX: это наиболее широко поддерживаемый формат данных GPS. *GPSBabel* имеет оболочку GUI, написанную на Qt. GUI строит командную строку, а та передается *GPSBabel*. Команда отображается



OpenStreetMap показывает всплывающую POI Эллы и маршрут навигатора для похода по Уэльсу.

в GUI; освоив программу, вы сможете использовать командную строку для загрузки маршрутов.

Многие современные устройства GPS имеют порты USB и опознаются как устройства массового хранения данных, так что вы сможете загрузить журнал с помощью файлового менеджера.



Определим допуски

Я опорожняю журнал из своего GPS-навигатора, только когда он заполнится. Однако я не хочу загружать весь журнал целиком после каждой пешей или велосипедной прогулки. *GPSPabel* позволяет выбрать различные фильтры, в том числе интервала времени, чтобы уточнить, какие данные загружаются из вашего навигатора. Следующая команда загружает журнал для указанной даты и сохраняет его в виде файла GPX:

```
gpsbabel -t -i garmin -f /dev/ttyUSB0 -x track, start=20120212000001, stop=20120212235959 -o gpx track.gpx
```

Вы – настоящий профи, и показания часов на вашем фотоаппарате совпадают с показаниями часов в устройстве GPS? Кхе... терпеть не могу в этом сознаваться, но у меня почему-то так не получается.

Часы в вашем навигаторе синхронизируются со спутником GPS, и время на них всегда точное.

«Чтобы показать, где сделаны фотографии, вы создаете POI.»

Почти все цифровые камеры хранят информацию о том, когда была создана фотография, в тэгах EXIF. Соответствующие тэги – 'DateTimeOriginal', 'CreateDate' и 'TimeCreated'. И все, что вам нужно

сделать – это обновить эти тэги, используя разницу во времени на камере и устройстве GPS.

Для уточнения информации EXIF мы воспользуемся инструментом *exiftool* от Фила Харви [Phil Harvey] (www.sno.phy.queensu.ca/~phil/exiftool). В моем случае часы на камере на 1 час 21 минуту опережали часы GPS, так что в тэгах EXIF мне надо было отнять 1 час 21 минуту от указанного там времени. С *exiftool*, команда такова:

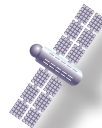
```
exiftool "-DateTimeOriginal=0:00:00 01:21:0" \
"-CreateDate=0:00:00 01:21:0" \
"-TimeCreated=0:00:00 01:21:0" *.JPG
```

Обратите внимание на – перед знаком равенства: он уменьшает значение времени. Перед обновлением тэгов EXIF создайте резервную копию фотографий, на случай ошибки.

Можно также создать копию данных EXIF в текстовом файле для всех фотографий в текущей директории, командой

```
exiftool -s *.JPG > exif_backup.txt
```

Если все получится криво, вы сможете использовать их для восстановления исходных данных.



Геотэги

Следующий этап – применить отметки времени EXIF в фотографиях для их корреляции с журналом навигатора. Затем мы запишем данные положения GPS в соответствующие тэги EXIF в фотографиях. Этот процесс известен как создание геотэгов.

Для создания геотэгов фотографий воспользуемся *GPSPPrune*. Если вы уже ликвидировали разницу во времени между вашими записями GPS и фотографиями, первым шагом будет загрузка маршрута GPS в *GPSPPrune*.

В меню Photo выберите Add Photos [добавить фото] и выберите директорию, где хранятся ва-

ши фотографии. В меню Photo выберите Correlate Photos [скоррелировать фото]. Множество опций в этом диалоговом окне позволят вам настроить Correlation Limits [допуски], поскольку есть вероятность, что данные в вашем журнале GPS и у фотографии в точности соответствуют друг другу. Две самых полезных опции –

- » Time Limit [допуск по времени]
- » Distance Limit [допуск по расстоянию]

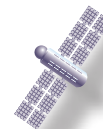
Если время или расстояние в поле EXIF и на трассе лежат в установленных вами допусках, то *GPSPPrune* будет считать, что фотография и точка в журнале совпадают.

Затем *GPSPPrune* отметит флажками все фото, где можно привести в соответствие время EXIF со временем GPS. Когда вы нажмете на OK, *GPSPPrune* отметит точки, соответствующие каждой фотографии, на вашей трассе. По нажатию на точку появится привязанная к ней фотография. После этого можно вписать широту и долготу в соответствующие поля EXIF каждой фотографии. Выберите Save to EXIF в меню Photo. Отобразится список фотографий, в которые запишутся данные EXIF.

В порядке меры безопасности, можете выбрать режим неперезаписи исходных фотографий – в этом случае они скопируются в файл с расширением **_original**.

Теперь *GPSPPrune* будет присоединять новые точки следования к концу вашего GPS-файла, с местоположением каждой фотографии, меткой времени и полным путем и именем файла фотографии. Пример приведен ниже:

```
<trkpt lat="53.06943558"
lon="-4.18059098"><ele>749</ ele><time>2011-02-20T09:53:57Z</time>
<link
href="/home/ian/Pictures/Holidays/North_Wales/2011-02-North_Wales/20022011085.jpg">
<text>20022011085.jpg</text>
</link></trkpt>
```



Создание миниатюр

Вам, естественно, захочется создать миниатюры для каждого изображения, которое вы намерены показывать на карте. Есть десятки способов сделать это, но мы воспользуемся простым скриптом оболочки, который вызывает программу конвертирования *imagemagick* для создания миниатюр шириной 75px. Миниатюры будут названы именем исходного файла, к которому припишется расширение **.thumb**.

Поместите следующий скрипт, который находится на DVD, в директорию с вашими фотографиями:

```
FILES="$@"
for i in $FILES
do
echo "Обработка изображения $i ..."
/usr/bin/convert -thumbnail 75 $i thumb.$i done
и запустите
./thumbnail.sh *.JPG
```

Из всех файлов с расширением .JPG в этой директории создадутся миниатюры.



Интересные места



Чтобы показать место, где были сделаны фотографии на вашей карте, вы создаете точку Point of Interest (POI), соответствующую каждой из них. К счастью, OpenLayers загрузит данные для отображения POI из файла, так что вам не придется добавлять все POI вручную. Поля записи данных показаны ниже. Учтите, что поля разделяются знаками табуляции, а записи – переводом строки.

```
lat lon title description icon iconSize iconOffset.
[Широта Долгота Название Описание РазмерЗначка СмещениеЗначка]
```

В текстовом поле можно разместить URL'ы. Мы будем использовать это для отображения миниатюры фотографии, ссылки на полноразмерную фотографию и текста описания.

Точки следования, содержащие данные о фото, извлекаются из файла GPX, который вы создали в процессе создания геотэгов фотографий с помощью *GPSPruner*.

GPSPruner весьма удобно прилагает эти данные к концу вашего журнала, и мы легко сможем их добыть с помощью текстового редактора. Скопируйте данные XML, не забыв включить все необходимые тэги, и вставьте их в шаблон, приведенный ниже. Обратите внимание, что шаблон содержит единственную точку следования, помещенную для демонстрации, каким должен быть правильный синтаксис. Сохраните данные в файл, например, **photos.gpx**.

```
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.0" creator="GpsPrune v13
activityworkshop.net"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns="http://www.topografix.com/GPX/1/0"
xsi:schemaLocation="http://www.topografix.com/
GPX/1/0 http://www.topografix.com/GPX/1/0/gpx.
xsd">
<name>Your Walk Name</name>
<desc>Export from GPSPruner</
desc><trk><name>Your Walk Description</
name><number>1</number>
```

```
<trkseg>
<trkpt lat="53.13602777777778" lon="
"-3.9945555555555554"><ele>575</ele><link
href="/.images/DSCF1869.JPG"><text>DSCF1869.
JPG</text></link></trkpt>
</trkseg>
</trk>
</gpx>
```

Теперь конвертируем фотоданные XML из созданного вами файла **photos.gpx** в разделенные знаками табуляции данные, используемые для отображения POI в OpenLayers. Поскольку данные – это просто XML, для конвертирования воспользуемся скриптом Python. Полностью его код приведен на нашем DVD. Запустите скрипт с DVD и перенаправьте выходные данные в файл: **./generate_poi.py > poi.txt**. Заметьте, что в конце файла должна быть пустая строка, иначе последний POI не будет отображаться. Поместите полученный файл (**poi.txt**) в ту самую директорию, где находится страница HTML, которая создает карту.



Отображение на карте

Теперь у вас есть фотографии с геотэгами, миниатюры и текстовый файл, содержащий информацию для отображения фотографий в виде POI на карте. Конечная стадия – создание страницы HTML, содержащей всю информацию, необходимую для загрузки данных карты и отображения маршрута и фотографий.

Вы можете использовать практически любой веб-сервер, поскольку работа ведется в браузере.

Для отображения карты и сопутствующей информации мы будем использовать OpenLayers, библиотеку JavaScript с открытым кодом для отображения картографических данных. Она работает с большинством современных браузеров, и она полностью клиентская, не требующая никаких специальных программ на сервере.

Используя OpenLayers, вы можете отображать ваши данные с помощью одного и того же JavaScript, используя карты от всех провайдеров, включая Google, Ordnance Survey и OpenStreetMap.

Однако если вы хотите использовать Google Maps или Ordnance Survey, вам нужно будет подписаться, чтобы получить ключ API. Внимательно прочитайте их условия соглашения, поскольку там есть ряд ограничений на использование данных. В противоположность им, OpenStreetMaps совершенно не имеет ограничений.

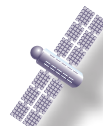
Выбор используемого сервиса зависит от ваших требований. В Великобритании я установил, что Ordnance Survey лучше всего подходит

для отображения пешеходных маршрутов, а OpenStreetMap – для велосипедных.

Впрочем, используя LayerSwitcher, виджет OpenLayers, вы сможете переключать свою карту между разными картографическими системами, и при желании использовать все три на одной и той же веб-странице.

Библиотека OpenLayers весьма обширна, и на первый взгляд кажется просто ошеломляющей. К счастью, в ней масса примеров, помогающих вам приступить к делу. Мы будем использовать созданный вами ранее файл GPX, чтобы отобразить маршрут на OpenStreetMap.

Карта также будет отображать POI, которые вы создали по тем местам, где снимали фотографии. При щелчке по POI отобразится всплывающее окно с подписью и миниатюрой фотографии. Если щелкнуться по миниатюре, во всплывающем окне отобразится полноразмерная фотография.



LayerSwitcher

Первый шаг – создать страницу, которая отображает карту интересующей вас местности. Будем использовать OpenLayers LayerSwitcher, который позволит переключаться между разными слоями OpenStreetMap. Вы также можете использовать ее для переключения между разными картами, например, Google и OpenStreetMap.

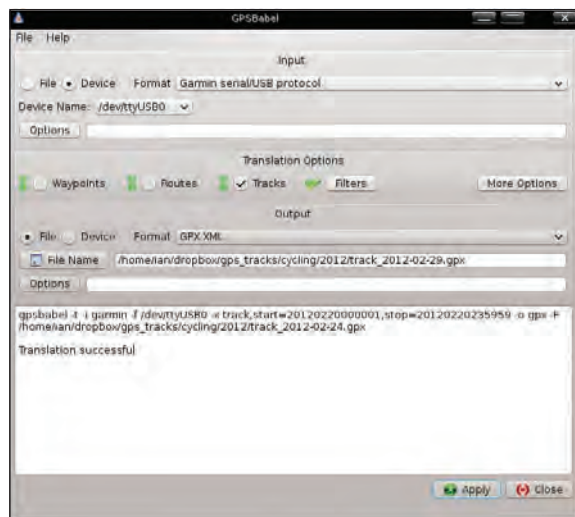
«Библиотека OpenLayers обширна и кажется ошеломляющей.»

Полный код для веб-страницы слишком велик, чтобы мы показывали его здесь, но он имеется на DVD. Вот несколько приемов и советов:

- » Обеспечьте, чтобы на вашей странице был действующий DOCTYPE, иначе начнутся странности.
- » Храните все файлы JavaScript на своем сервере.
- » Однако у вас будут все самые свежие заплатки (и новые ошибки тоже), если вы обратитесь к библиотеке на удаленном сайте.
- » Используя Google Maps или Ordnance Survey, убедитесь, что у вас действующий ключ API. Ключ привязан к вашему доменному имени, и на другом сайте работать не будет.
- » Слой Cycle Map [велосипедный] в OpenStreetMap также отлично подходит для отображения данных по пешеходным маршрутам, поскольку отображает контурные линии.

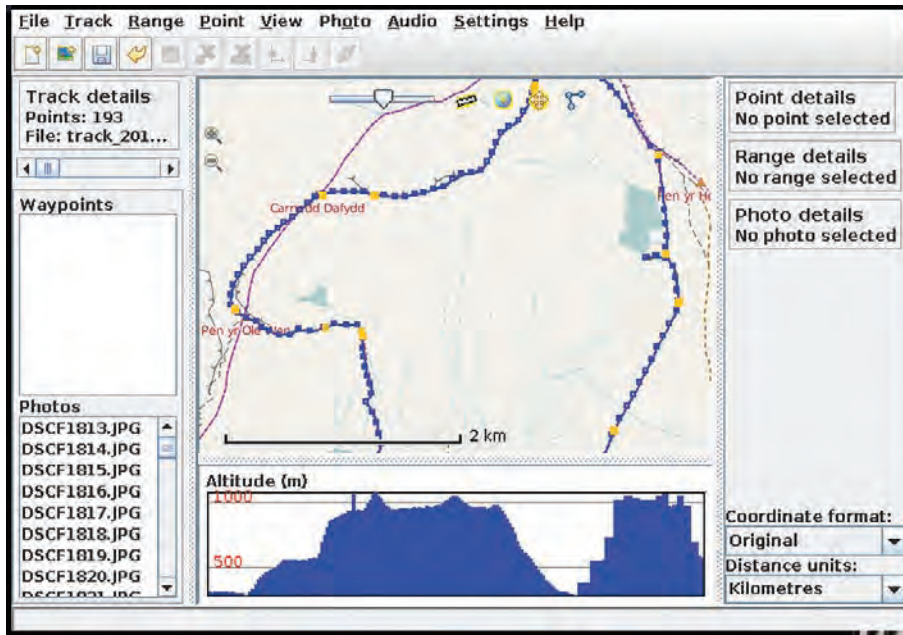
Код, отображающий слой карты, находится в одном блоке JavaScript. Сначала вы объявляете несколько переменных, которые сообщают карте нужную широту и долготу. Прочие атрибуты, например, отображение панелей LayerSwitcher и Pan [панорама] and Zoom [масштабирование], также должны быть определены.

Теперь добавьте слой, который отображает ваш маршрут GPX на карте в собственном слое. 'Carneddau' – название маршрута, который отображается в LayerSwitcher.



» Обработка файла с помощью *GPSBabel*.





➤ GPSPRUNE обрабатывает ваши маршруты GPS и наносит фото на трассу. При щелчке по точке на карте отобразится соответствующий пейзаж.

Если вы хотите отобразить более одного маршрута, просто скопируйте этот раздел и затем соответствующим образом измените параметры.

```
// Добавим слой с трассой GPX
var lgpx = new OpenLayers.Layer.
GML("Carneddau", "carneddau.gpx", {
format: OpenLayers.Format.GPX,
style: {strokeColor: "green", strokeWidth: 5,
strokeOpacity: 0.5},
projection: map.displayProjection});
map.addLayer(lgpx);
```

Далее, определите слой, который будет отображать ваши POI:

```
// Load our Points of Interest
var pois = new OpenLayers.Layer.Text("POI",{
location:"poi.txt",
projection: map.
displayProjection });
map.addLayer(pois);
```

И, наконец, можно добавить маркер, чтобы показать, где вы начали и где завершили маршрут.

```
// Добавим маркет начального пункта.
var size = new OpenLayers.Size(21, 25);
var offset = new OpenLayers.Pixel(-size.w/2,
-size.h);
var icon = new OpenLayers.Icon('http://www.
openstreetmap.org/openlayers/img/marker.
png',size,offset);
layerMarkers.addMarker(new OpenLayers.
Marker(lonLat,icon));
```

Вы должны определить `div` в теле страницы, на которой хотите отображать карту. Помните, что имя `div` то же, что и у слоя, использованного для определения карты – проявив недюжинную изобретательность, я назвал его 'map':

```
<div id="map" style="width: 800px; height:600px;
position: fixed;">
```

После загрузки страницы функция `body.onload()` вызовет функцию JavaScript `init()`, которая загружает данные карты. При щелчке на мини-

туре, отображенной в POI, полномерная фотография отображается во всплывающем окне.

Всплывающее окно создается с помощью JavaScript, который использует URL, заключенный в текстовом файле с данными для POI.

Графики восхождения

Мне часто хочется изобразить свой маршрут GPX в виде графика восхождения, который отображает высоту на всей протяженности маршрута. Если, как мы сказали, *GPSPRUNE* – это швейцарский нож для данных GPS, то для графиков этого звания достоин *Gnuplot*. Вы можете создать график с помощью *GPSPRUNE* (который вызывает *Gnuplot*), но работа непосредственно с *Gnuplot* будет более гибкой.

Мы воспользуемся скриптом Python `gpxplot` от Сергея Астанина [Sergey Astanin], но автор сделал свое ответвление, добавив функциональности. Эта версия имеется на Github по адресу <https://github.com/geekinthesticks/GPX-Tools>. Скрипт считывает данные из файла GPX и умеет давать на выходе графики в разных форматах. Запустите `.gpxplot.py -help`, чтобы познакомиться с опциями. Вам также нужно будет установить `gnuplot.py`, который имеется в большинстве дистрибутивов. Если у вас не установлен `gnuplot.py`, использование опции `--gprint option` сгенерирует скрипт *Gnuplot*, который можно передать напрямую *Gnuplot* из командной строки.

Вот каким образом на экране создается график маршрута с указанием высот:

```
./gpxplot.py --output-format gnuplot
--y-axis elevation \ --x-axis distance
--tznme 'Europe/
London' \
```

```
--imperial -file
arneddau.gpx
```

По умолчанию данные выводятся в метрической системе (высота – в метрах, расстояние – в километрах); опция `--imperial` отобразит высоту в футах, а расстояние в милях.

Если вы хотите вывести график в файл изображения, а не на экран, используйте опцию `--image`, за которой должно следовать имя файла. Формат изображения определяется расширением имени файла: так, `--image track.png` создаст файл PNG. Другие поддерживаемые форматы вывода – `GoogleChart`, `Gprint`, `table` (колонки, разделенные пробелом) и `orgtable` (таблицы в формате `orgmode Emacs`).

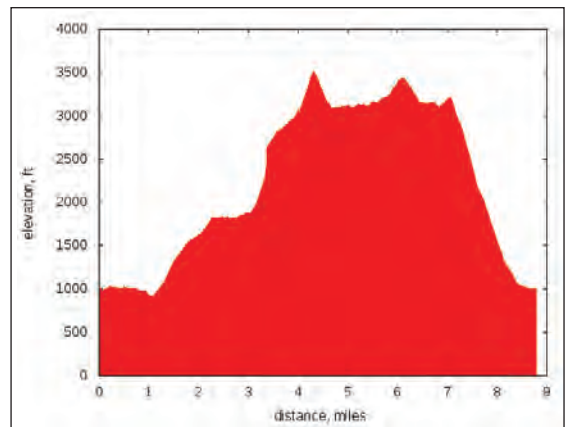
Время в файлах GPX обычно отображается в формате UTC. Желая отображать ваши данные по местному времени, используйте опцию `--tznme` и задайте часовой пояс в формате 'Europe/London'.

В *Gnuplot* масса опций для настройки цветов, заголовков и прочих параметров. Если вы решите подправить выходные данные, создайте файл *Gprint* и отредактируйте его, чтобы получить желаемый формат. Файл *Gprint* можно затем использовать в качестве исходной информации для *Gnuplot* при создании графика.

Заключение

На нашем уроке мы рассмотрели только основы – вы можете сделать намного больше, например, добавить слои карт для *Ordnance Survey* и *Google Maps*, чтобы ваши пользователи могли переключаться между разными картами.

Помимо всего кода из этой статьи с DVD, вы также можете получать обновления из репозитория автора на Github по адресу: <https://github.com/geekinthesticks/Plotting-gpx-Data-and-Showing-Geotagged-Photos-Using-OpenLayers>. LXF



➤ Создание графиков восхождения для маршрутов ваших прогулок вроде этого – простое и быстрое упражнение.



Битва мобильных титанов: Android и iOS

Какая мобильная ОС проще в использовании? Какая более функциональна? **Майк Сондерс** провел опыты на двух морских свинках...

Мы не устаем поражаться, сколько человек использует Linux, не подозревая об этом. Время от времени мы сталкиваемся с теми, кто уверяет, что Linux слишком сложен или разработан только для технарей и серверов, а у самих в карманах болтается Android.

Конечно, Android сильно отличается от обычного настольного дистрибутива Linux, однако ядро-то у него пингвиное, и его популярность быстро растет. Но у него есть

лукавый враг – iOS. Мобильная ОС от Apple, применяемая на iPhone, iPod Touch и iPad, имеет репутацию чрезвычайно дружелюбной к новичкам.

«Мобильная ОС от Apple имеет репутацию дружелюбной к новичкам.»

В то же время Apple критикуют за чрезмерную ограниченность ее продуктов, которые заранее объ-

являют «способ Apple [Apple way]» наилучшим и не дают пользователям воли. Об этом высказывается тысяча и одно мнение, и мы решили исследовать вопрос снаружи – на людях, у кого нет особой приверженности ни к Linux, ни к Apple, или просто тех, кто предпочитает выбирать лучший продукт.

Итак, мы нашли двух безразличных к ОС субъектов исследования и провели с ними серию опытов. **Барбара** [Barbara] – новичок в смартфонах, она перешла

на Samsung Galaxy (который работает под Android) за неделю до написания статьи. Она начала в нем осваиваться, но с iOS в жизни не общалась, и мы попросили ее проделать задания на iPhone. У Сабы [Saba] – обычный телефон Sony Ericsson, а вот смартфона у нее никогда не было. Зато у нее есть iPod Touch, она знакома с iOS, но раньше никогда не использовала Android, поэтому выполняла задания на Galaxy Барбары.

В порядке вводки мы попросили участниц описать их нынешние сложившиеся впечатления об обеих платформах.

«С телефонами Android трудно понять, какой хорош, а какой плох, в магазинах их очень много», размышляет Саба. Но она и не в большом восторге от процесса установки на устройствах Apple: «Очень нудно использовать iTunes для всего подряд, чтобы активировать и качать музыку и фильмы на мой iPod». Apple предпринял ряд шагов, чтобы освободить устройства с iOS от этого монстра iTunes, однако здесь еще есть куда двигаться.

У Барбары свой подход: «iPhone есть у всех, и приятно быть непохожим на других. В Android выбора больше».

Интернет

Нашей первой задачей было подключиться к беспроводному интернету. Саба споткнулась сразу же, зайдя в Galaxy в Settings > Wireless и увидев

«Сходство объясняет, почему компании угрожают друг другу судом.»

огромный список беспроводных сетей, сохранившихся от предыдущих соединений, большинство которых уже были недоступны. «А к какой подключаться?» – постоянно спрашивала она, прокручивая список. Барбара в iPhone зашла в Settings > Wi-Fi прямо из домашнего экрана, и была более успешна.

Затем мы попытались перейти на сайт и сделать закладку на него. Сразу стало понятно, что обилие физических кнопок на Galaxy смущает Сабу, в особенности кнопка Back. «Я просто привыкла, что есть одна кнопка, чтобы попасть на домашнюю страницу», сказала она о своем iPod, но Барбара возразила: «Она очень удобная, ты всегда можешь вернуться на предыдущий шаг, не переходя на стартовую страницу.»

Итого: одно очко для iOS за простоту, и одно для Android за гибкость.

Приложения

Затем мы стали пробовать встроенные приложения. Очень интересно наблюдать сближение этих двух платформ, поскольку большая часть программ, предлагаемых рынком, ведут себя почти одинаково. Такое сходство объясняет, почему сейчас все компании угрожают друг другу судом, хотя мы и не уверены, что это приведет к хорошей конкуренции и лучшим продуктам... Как бы то ни было, одним из наших заданий было до-

бавление нового контакта; и именно здесь стало проявляться фундаментальное различие между двумя платформами. Барбара ткнула на Contacts на экране iPhone, нажала кнопку с плюсом и ввела данные. Саба попыталась сделать то же самое, но у нее возник дополнительный шаг: куда сохранить новый контакт – на SIM-карту или в память телефона. «Что это значит? – негодовала Саба, – не все ли мне равно?» Однако Барбара была более позитивна: «Тебе, может, и без разницы, а кому-то это важно».

Мы попробовали сделать снимок и удалить его. У Барбары с iPhone не было проблем, но Саба застопорилась на удалении, а именно – не сумела отыскать только что сделанную фотографию без возврата на домашний экран и потом на всю галерею. Кто виноват? При создании фотографии в iOS недавнее фото отображается в виде миниатюры в углу, и вы знаете, как перейти к нему одним прикосновением. В Galaxy есть только кнопка со стрелкой, и вовсе не очевидно, что она ведет к нужному фото.

Изменение настроек

Вызвать правильный диалог настроек – трудная задача в любой операционной системе, а для мобильной платформы это еще сложнее. Для начала мы попросили обеих девушек сменить картинку на домашнем экране. Барбара пошла прямо к значку Settings и нашла там опцию Wallpaper.

Саба тоже перешла в Settings, в подменю Display, однако потратила целую минуту на поиск хоть чего-нибудь про обои. В конце концов она сдалась, вернулась на домашний экран, запустила просмотрщик фото и задала новые обои с его помощью. Затем мы подсунили задание посложнее: поменять клавиатуру и вставить спецсимволы (например, буквы с диакритическими знаками и ударениями).

«В первые пару дней я не знала, что можно печатать различные символы, просто удерживая клавишу, – объяснила Барбара. – Это вышло случайно». У нее не было проблем с выполнением этой задачи на iPhone, так как она уже освоила этот трюк. «Но я не знаю, как это можно постичь обычным способом» – это кажется нам справедливым: ну кто сейчас читает инструкции пользователя? И почему это должно быть спрятано в долгих нажатиях клавиш (или в случайных нажатиях, вроде удерживания Alt для выключения компьютера? Здесь мы смотрим на вас, дизайнеры Gnome 3).

Саба сделала на своем iPod такое же открытие, и знала, что нужно делать на Android, но здесь мы столкнулись с потенциальной проблемой. Когда в iPhone выбрана английская клавиатура, вы можете, например, удерживать клавишу U и видеть разные опции для других языков – умляути, акценты и прочее. Удерживайте S – и вы увидите немецкую ß и используемую в словацком языке Š. А в Galaxy Сабе приходилось использовать одну клавиатуру для немецких символов, и другую – для словацких.



» В наших высоконаучных и всеобъемлющих испытаниях Android и iOS идут голова к голове.

Окончательные мнения

Итак, после часа исследований двух платформ, к каким выводам пришли субъекты исследования?

Барбара: «iPhone выглядит как урезанная версия Galaxy. Его легко использовать, но иногда остается чувство пустоты, то есть как будто многих опций не хватает. Остается ощущение, что для каждого действия есть только один способ, а в Android таких способов много».

Саба: «Я думала, что Android будет сложным, но он оказался почти как мой iPod. Иногда нелегко отыскать то или иное, и возникает больше вопросов, чем с iPod, но к этому можно привыкнуть. И я уже почти освоилась с кнопкой Back».

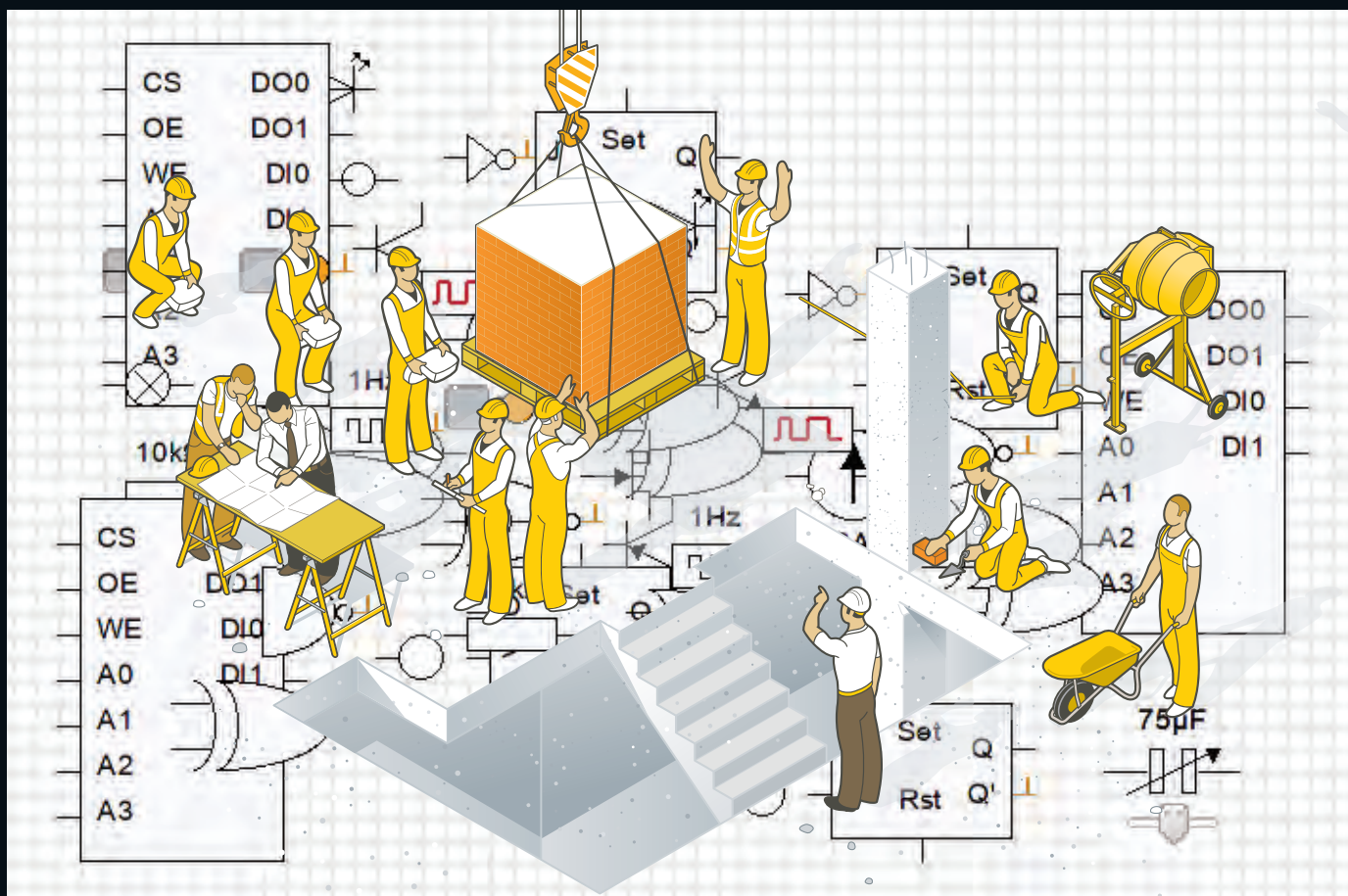
» Простота: iOS

» Функциональность: Android

Это волнует, поскольку, по нашему ощущению, Android может постепенно перенять простоту iOS, не жертвуя гибкостью и функциональностью. Существуют обычные проблемы и камни преткновения, и если их сгладить, то Android будет состязаться с предложениями для новичков от Apple. Однако Apple опасается открывать iOS, и Android всегда будет иметь преимущество в гибкости, функциональности, настройках и загрузке некоммерческих приложений. И, конечно, Linux сам по себе удивителен. Сейчас iOS, быть может, обгоняет с точки зрения простоты и глянца, однако Android побивает своим набором возможностей. Прекрасные времена для поклонников Linux. LXF



» Сколько ни объясняй мне, Барбара, я все равно не вижу смысла в Angry Birds.



Компьютер своими руками



Узнайте, как бьется электронное сердце вашего компьютера, вместе с **Джонатаном Робертсом**.

А вы когда-нибудь задумывались о том, как работают компьютеры? Задумайтесь. Ведь это же обычные электронные устройства, вроде лампочки, только из инертных металлических и кремниевых деталек.

Но, будучи правильно собранными, эти крошечные компоненты способны хранить информацию и управлять ею – и даже преобразо-

вывать наш мир. Вот о чем мы задумались в нашей дозорной башне LXF, и решили как следует в этом разобраться. Теперь же спешим поделиться с вами всем, что нам удалось выяснить. А с помо-

«Правильно собранные, эти компоненты способны преобразовывать мир.»

щью свободной программы *KTechlab* – еще и покажем, как применить это знание на практике, дабы создать свое собственное, пусть и примитивное, вычислительное (ну, или счетное) устройство.

Если вас это заинтересовало и вы решили создать компьютер своими руками, все, что вам нужно – это макетная плата и кое-какие компоненты. Можно заглянуть в каталоги электроники Marlin, например.

Переключение

Для начала припомним, как работает лампочка. Тут все просто – щелкаете выключателем, и она зажигается. Щелкаете еще раз – и она гаснет. Этим действием вы каждый раз создаете и разрываете электрическую цепь – контролируете поток электроэнергии.

По тому же принципу работает и компьютер. Огромное множество (миллиарды) крохотных переключателей точно так же управляют током. Единственное отличие, что в данном случае это не механические переключатели, управляемые вручную, а электрические, управляемые самой электроэнергией!

Звучит странно, но справа перед вами пример такого устройства, называемого реле (рис. 1). Это электрическая цепь, представляющая собой лампочку, соединенную с батарейкой, а между ними – выключатель. Пока он открыт, цепь разорвана, и лампочка не горит.

Однако под выключателем помещен электромагнит. Это прибор, проявляющий свои магнитные свойства при прохождении через него тока. Когда электромагнит включен, выключатель замыкается притяжением магнитного поля, и лампочка загорается. А если его выключить, выключатель вновь размыкается, и лампочка гаснет.

ОК, ловлю вас на мысли, что «все равно кто-то должен включать электромагнит». Верно. И этим мы еще займемся, но пока давайте посмотрим, каким образом работа переключателя может зависеть от наличия электрического тока.

Электрические переключатели бывают устроены по-разному. Это может быть реле, радиолампа... наконец, транзистор. Последние – самые компактные, поэтому в современных компьютерах используются именно они.

Мы не будем объяснять, как они работают, чтобы не погружаться в дебри физики; поговорим лучше об их применении, ведь они будут основой схем, описываемых нами далее в этой статье.

Транзистор как выключатель

Чтобы продемонстрировать вам работу транзистора, покажем, как он используется в качестве переключателя. Посмотрите на рис. 2.

Перед вами вновь схема, почти такая же, как и в примере с реле. Только теперь вместо реле используется транзистор. Две ножки транзистора подцеплены так же, как и внутренний переключатель реле, а третья присоединена к управляющему проводу.

Когда небольшой ток подается на третью ножку (называемую базой), основной ток практически без потерь течет между двумя другими (та, что со стрелкой, называется эмиттер, другая – коллектор). Соответственно, если напряжение на базу не подается, передачи тока между ними тоже нет.

Булева логика

Супер! Теперь вы знаете, как работает транзистор, но вот зачем он нам? Чтобы в этом разобраться, нам придется вернуться немного назад и поговорить о булевой логике.

Булева логика – это система определения значений различных высказываний, примерно как в обычной математике. Но значений в этой системе существует только два: 1 или 0 – истина или ложь.

Для примера применим булеву логику к предложению «У меня холодные руки». Если они

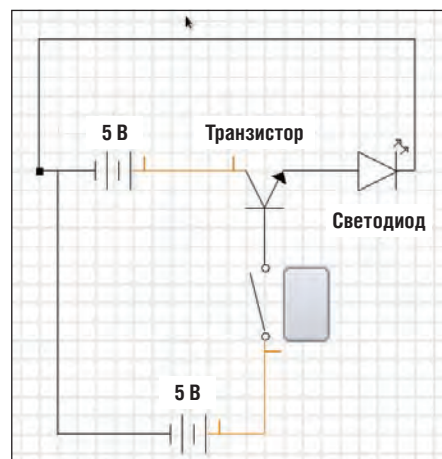


Рис. 2. Простой способ использования транзистора как переключателя.

действительно холодные, это истина. Если нет – ложь. Однако булева логика не сводится к оценке единичного высказывания, поскольку, как в любой математике, здесь есть операторы.

«У меня холодные руки И теплые ноги» будет истинным, только если обе части выражения вер-

«Чтобы разобраться, нам придется поговорить о булевой логике.»

ны; «У меня холодные руки ИЛИ теплые ноги» – если верна хотя бы одна часть, или обе.

Подобные высказывания часто упорядочиваются при помощи таблиц истинности, и ниже вы видите ее для рассмотренных нами предложений:

Высказывание 1	Высказывание 2	Результат И	РЕЗУЛЬТАТ ИЛИ
Ложь	Ложь	Ложь	Ложь
Истина	Ложь	Ложь	Истина
Ложь	Истина	Ложь	Истина
Истина	Истина	Истина	Истина

Есть еще одна ключевая операция, с которой мы столкнемся – операция НЕ. Она интерпретирует единичное высказывание и меняет его значение на противоположное. Таблица истинности будет выглядеть так:

Высказывание 1	Результат НЕ
Ложь	Истина
Истина	Ложь

Логика и транзисторы

Оказывается, способность оценить истинность или ложность высказывания при помощи операторов И, ИЛИ и НЕ лежит в основе работы компьютера. Почему это так, вы поймете чуть дальше, »

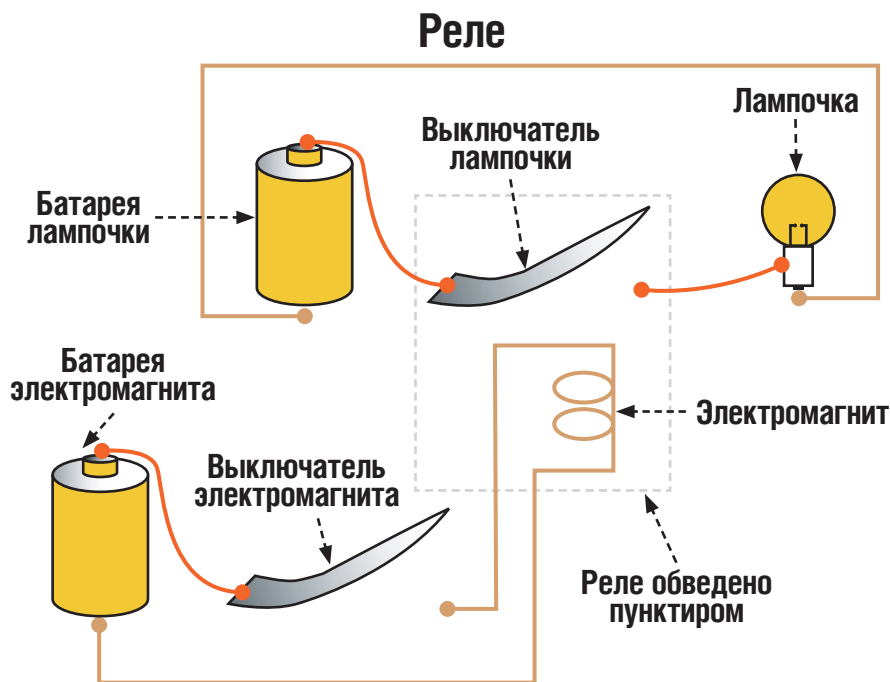


Рис. 1. Если электромагнит включен, выключатель лампочки замыкается; если магнит отключить, выключатель вновь откроется, и лампочка погаснет.

а сейчас посмотрим, каким образом скромный транзистор позволяет применять булеву логику.

Во-первых, отметим, что, в отличие от приведенных выше высказываний, в случае с электрической цепью у нас есть только Вкл или Выкл, как и с лампочкой. К счастью, булева логика работает как раз с двумя значениями. Итак, теперь 1 (Вкл) будет обозначать истину, а 0 (Выкл) – ложь.

Для начала создадим так называемый Вентиль И – то есть эквивалент булевой операции И в электронике. Обратимся к рис. 3. Перед вами вновь цепь. На этот раз без батареи, но с источником постоянного напряжения. Чтобы замкнуть цепь, нам требуется заземление, которое будет играть роль второго контакта в батарее. При наличии разрыва между ними ток протекать не будет.

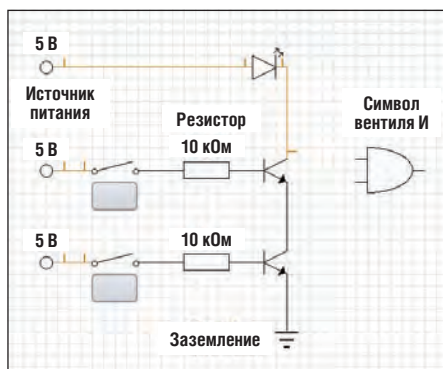
Светодиоды и транзисторы

Имейте в виду, что светодиод используется как индикатор результата нашей булевой операции, и что два резистора (маленькие детали, уменьшающие силу тока – что можно сделать также и понизив напряжение в устройстве) нужны для защиты транзисторов. Иначе результаты могут оказаться весьма неожиданными!

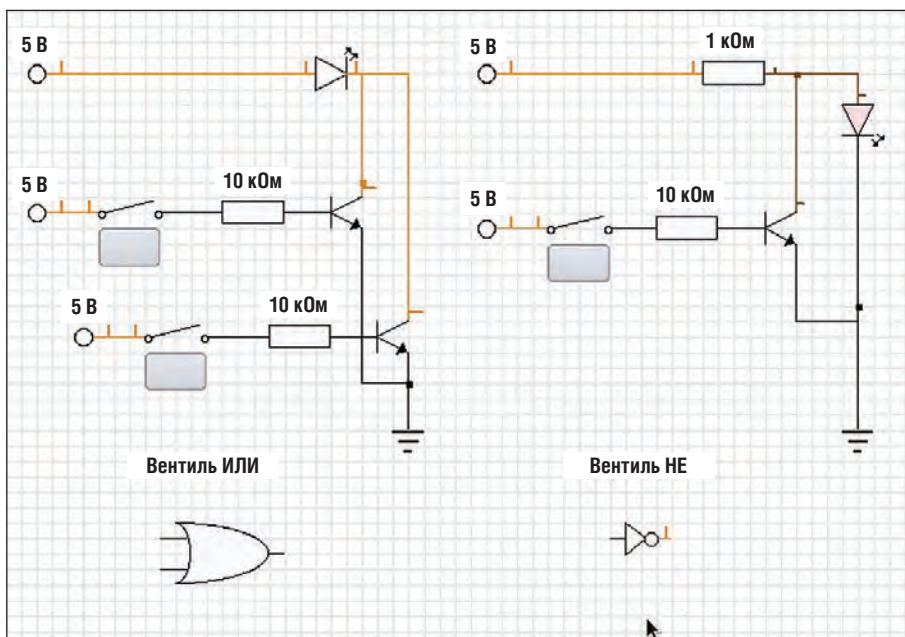
Теперь взгляните на транзисторы. Они вновь используются как переключатели. Что интересно, они оба расположены на одной линии между светодиодом и землей. Чтобы замкнуть цепь (тогда светодиод зажжется, и полученный результат будет «истина»), оба транзистора должны пропускать ток от коллектора к эмиттеру. Если любой из них разомкнут, тока в цепи не будет, и результат будет «ложь».

Если хотите попрактиковаться с такой цепочкой самостоятельно и убедиться, что она на самом деле воспроизводит таблицу истинности для И, можете установить *KTechlab*. Поперетаскивайте там компоненты с левой стороны на правую, поперебрасывайте провода и пощелкайте переключателями реле, чтобы увидеть результат. Любую цепь, о которых пойдет речь далее, можно воссоздать в *KTechlab*, а на нашем DVD вы найдете необходимые файлы в формате **.circuit**.

Иллюстрация человека в желтых комбинезонах, толкающего желтую тележку.



➤ Рис. 3. Эта схема – пример применения переключателя для выполнения логической операции И.



➤ Рис. 4. Вентили ИЛИ и НЕ плюс вентиль И с рис. 3 – все, что нужно для изготовления действующего компьютера.

Еще два вентилля

Прежде чем двигаться дальше, давайте посмотрим, как на транзисторах работают вентили ИЛИ и НЕ, поскольку все остальное составляется при помощи этих трех компонентов. Схемы обеих цепей представлены на рис. 4.

Начнем с элемента ИЛИ, он проще. Тут все почти так же, как и в случае с И, но вместо подключения транзисторов последовательно мы теперь подсоединим их к источнику питания и проводу заземления параллельно. Тогда электроток не будет, если оба они отключены, но если хотя бы один включен, то светодиод загорится.

С элементом НЕ будет немного сложнее. Свет уже горит по умолчанию, поскольку светодиод подсоединен напрямую между питанием и заземлением. Это отвечает первой части таблицы истинности. Но почему, если подключить транзистор, свет погаснет? Ведь порядок подключения не изменился?

Причина в том, что в параллельной цепи, подобие этой, ток распределяется между двумя ветвями (в данном случае, между транзистором и светодиодом) в соответствии с их сопротивлением. Поскольку у транзистора сопротивление низкое, почти весь ток пойдет через него, а не через светодиод.

Конечно, это не идеально, и чтобы обеспечить недостаточность силы тока для загорания светодиода, мы поставили резистор, дополнительно рассеивающий энергию.

Простая логика

Вот, собственно, и все, что нам нужно знать о транзисторах. С помощью этих трех операций можно создать компьютер! Просто не верится. Построение более сложных схем на их основе может показаться запутанным, но, к счастью, *KTechlab* уже содержит готовые вентили И, ИЛИ

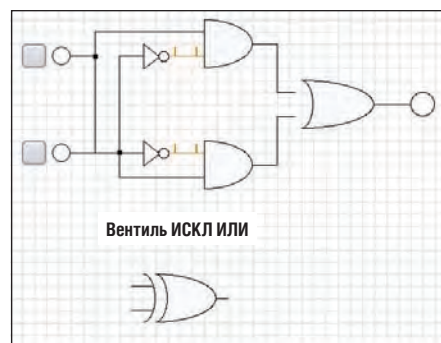
и НЕ, с четкими, читаемыми обозначениями, что значительно упрощает работу. Их-то мы и будем использовать.

Символ каждого элемента соответствует схемам, которые мы показывали на рисунках. Как можно совместить их воедино, демонстрирует рис. 5. Здесь представлен элемент ИСКЛ ИЛИ (Исключающее ИЛИ – истинно, только если активен один из вводов, но не оба сразу), собранный из уже составленных нами элементов. Его мы тоже будем впоследствии использовать, поэтому обратите внимание на соответствующий ему символ на рис. 5.

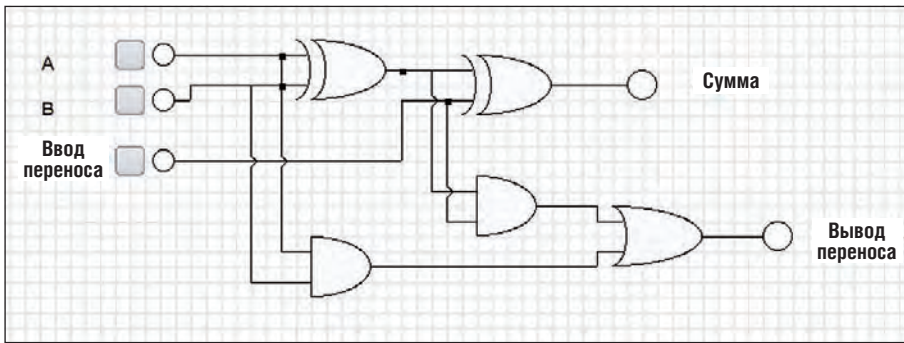
Двоичная математика

Ну вот, мы накануне изготовления своей собственной счетной машины; но прежде чем ее делать, разберемся в двоичной системе счисления.

Числа можно представлять по-разному. В повседневной жизни мы чаще всего используем десятичную систему. В ней 10 символов, значение



➤ Рис. 5. Здесь, при помощи готовых вентилей И и ИЛИ из *KTechlab*, мы составили элемент ИСКЛ ИЛИ. Для удобства восприятия мы используем уже не светодиоды и источники питания, а логические операции ввода и вывода.



► Рис. 6. Полный сумматор может складывать любые многозначные числа, просто передавая Вывод переноса с одного на Ввод переноса другого.

которых зависит от позиции: каждая цифра значит в 10 раз больше стоящей справа.

Например, в числе 133 крайняя правая 3 – это просто 3. А вот вторая 3 – это уже 30; ну, а крайняя левая единица обозначает 100. Значение самого числа складывается из суммы всех трех позиций.

В двоичной математике все немного по-другому. Вместо 10 символов – только два: 0 или 1. И поскольку их всего два, каждая позиция двоичного числа в два раза больше того, что стоит справа от него, а не в десять раз.

Рассмотрим, например, двоичное число 011. Единица справа равна 1, а следующая – уже два.

«Двоичная система удобна для компьютеров, поскольку более проста.»

И, как и прежде, значение числа – это сумма обеих позиций, поэтому 011 в двоичной системе – это 3 в десятичной.

Отметим, что двоичная система отличается лишь представлением самих чисел, но работает точно так же. То есть привычные функции сложения, вычитания и т.д. в ней тоже применимы.

Двоичная система удобна для компьютеров, поскольку более проста в использовании и хранении на электронных устройствах. 1 в двоичной системе может обозначать наличие тока (закрытый переключатель), а 0 – его отсутствие (открытый переключатель).

Полусумматор

Если вам кажется, что к такой системе применима булева логика, вы совершенно правы. И далее мы совместим все, что мы узнали, чтобы создать простую машину, способную складывать двоичные числа.

Сначала представим, что нам нужно для сложения двух двоичных чисел. Попробуем выписать двоичные суммы для всех возможных комбинаций 0 и 1. Можно считать столбиком и запоминать остаток, как в простейшей арифметике.

Вы убедитесь, что поскольку для сложения 1+1 недостаточно одного двоичного разряда, нам потребуется два. Можно представить это в виде таблицы истинности:

A	B	Сумма	Перенос
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Посмотрите на значения колонок в таблице истинности. Колонка Перенос выглядит так же, как и таблица истинности элемента И, в то время как Сумма совпадает с ИСКЛ ИЛИ (но не ИЛИ, поскольку вывод не может быть 1, если оба ввода равны 1).

Эти правила мы можем представить в виде схемы, подсоединив вентиль И и вентиль ИСКЛ ИЛИ к одним и тем же вводам. Вы-

вод И представляет перенос, вторую цифру в результате сложения, а ИСКЛ ИЛИ – сумму, первую цифру. Соответствующую схему вы увидите в файлах на диске.

Полный сумматор

То, что мы рассмотрели, называется полусумматор. Круто звучит, да? Теперь вы видите, как при помощи тока можно заставить что-либо работать – нам удалось произвести математическую операцию, просто направляя электроны в разные стороны!

Однако с полусумматором мы далеко не уйдем. Ведь он поддерживает только операции с двузначными числами, и мы не можем применить его к числам больше 11 (3, в десятичной системе). Для этого есть полный сумматор, и, как следует из названия, он состоит из двух полусумматоров (с небольшими изменениями).

Принцип тот же, только на сей раз мы складываем два двузначных двоичных числа. Если продолжать аналогию со сложением столбиком, это все равно что посчитать две однозначных суммы. В одной колонке складываете, запоминаете перенос, если нужно, а затем считаете сумму в другой колонке и опять же запоминаете перенос.

Таким образом, сложение сколь угодно больших чисел превращается «просто» в многократное сложение единичных чисел, множеством полусумматоров. Мы говорим «просто», потому что во вто-

рой колонке вы можете увидеть три ввода: A, B и перенос от предыдущей операции. Чтобы разобраться, как именно это получается, составим еще одну таблицу истинности:

A	B	Ввод переноса	Сумма	Вывод переноса
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

Внимательно изучите эту таблицу – попытайтесь понять сначала, какими должны быть выходы, чтобы прийти к колонке Сумма, а затем что нужно, чтобы прийти к Вывод переноса. Вы увидите, что Сумма = (A ИСКЛ ИЛИ B) ИСКЛ ИЛИ C и Вывод переноса = (A И B) ИЛИ (Ввод переноса И (A ИСКЛ ИЛИ B)).

Представив это в виде схемы, как на рис. 6, вы увидите, что в результате получается две полусуммы, с дополнительным вводом, а их переносы объединяются элементом ИЛИ.

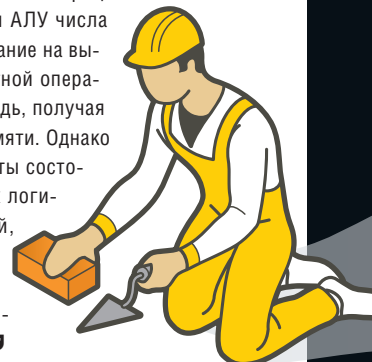
Вот что такое полный сумматор. Сам по себе, он может складывать три однозначных числа, но «каскад» таких сумматоров при подаче Вывода переноса одного полного сумматора на Ввод переноса другого делает возможным сложение любых чисел. Каждый полный сумматор в этом случае будет представлять «одну колонку» итогового суммы.

Начало большого пути

Вот, собственно, и все. Мы прошли долгий путь от простейших электрических цепей до счетной машины, способной складывать любые многозначные числа. Но нам еще далеко до создания реального компьютера.

То, что мы создали в данной статье, могло бы быть частью компьютерного арифметическо-логического устройства. В настоящей машине АЛУ выполняло бы и многие другие операции, включая вычитание, умножение, а также простые логические операции, такие как И, ИЛИ, сдвиг влево и вправо. Центральный процессор поставил бы АЛУ числа

для расчета и задание на выполнение конкретной операции, в свою очередь, получая эти данные из памяти. Однако все эти компоненты состоят из простейших логических действий, рассмотренных выше, только по-разному скомбинированных. LXF





MINIX 3

Вторая жизнь первопредка

Натешившись историей, **Алексей Федорчук** начинает задумываться о будущем – не будет ли оно принадлежать микроядрам?

Кто же из линуксоидов не знает старика MINIX'а? Эта минорная учебно-показательная UNIX-подобная ОС была создана четверть века назад профессором Эндрю Таненбаумом [Andrew S. Tanenbaum]. Предназначалась она для вразумления студентов и приобщения их к идеалам UNIX на самой демократической платформе всех времен и народов – на IBM PC-совместимых компьютерах. Она уже фигурировала в нашей истории – в статье о зарождении Linux (**LXF146**, июль 2011 г.). Ибо именно MINIX вразумил Линуса Торвалдса настолько, что он занялся сочинением своей терминальной программы, которой суждено было превратиться в Linux.

В той же статье была описана и история ОС MINIX, сначала 1, а потом и 2, которую можно рассматривать как предысторию героини сегодняшнего рассказа. И которая завершилась анонсом MINIX 3, состоявшимся 24 октября 2005 года. Несколько повторысь, подчеркну: это была не просто следующий релиз прежних MINIX'ов, а совершенно новая ОС, и цифра 3 здесь – не номер версии, а часть ее имени собственного. Таненбаум неоднократно подчеркивал, что сходство ее с предшественниками – лишь в первом компоненте названия, а различие между MINIX 1/2 и MINIX 3 не меньше, чем между Windows 3.1 и Windows XP.

MINIX 3 более не рассматривалась как учебно-показательная разработка, а позиционировалась как всамделишная операционная система общего назначения, предназначенная, в перспективе, для широкого класса устройств, в том числе и встраиваемых. Это символизировалось и сменой правового статуса системы: отныне она распространялась под лицензией BSD.

Официальный сайт проекта – <http://www.minix3.org/>. Интересно, что буквально через несколько месяцев после анонса MINIX 3, 1 февраля 2006 года, Романом Игнатовым был создан русскоязычный ресурс по этой системе – <http://www.minix3.ru/>, который успешно развивается и по сей день.

Отличие новой системы заключалось еще и в модели разработки. MINIX 1 и MINIX 2 были фактически личными творениями Эндрю Таненбаума, а все дополнения к ней, вроде ставшего знаменитым патча Брюса Эванса [Bruce Evans] (именно с его дополнением использовал MINIX Линус Торвалдс для работы над своей будущей операционкой), носили, в силу условий распространения, сугубо неофициальный характер.

К разработке же MINIX 3 Таненбаум с самого начала привлек широкий круг участников – от своего соавтора по второму изданию учебника Альберта Вудхалла [Albert Woodhall] до студентов, аспирантов и сотрудников Университета Вреие [Vrije Universiteit], а также волонтеров. Состав участников проекта, по понятным причинам, был весьма текучим, и перечислить их всех поименно не представляется возможным. Актуальный на данный момент список можно видеть здесь: <http://wiki.minix3.org/en/WhosWorkingOnWhat>. Однако нельзя не отметить среди них наших соотечественников – упомянутого выше Романа Игнатова и Евгения Иванова.

О микроядрах

В MINIX 3 воплотилось представление Таненбаума и его соратников о том, какова должна быть современная операционная система. Однако, чтобы говорить о нем, надо опять обратиться к истории – на сей раз к истории микроядерных ОС (краем этот вопрос был затронут в предыдущей статье цикла, см. **LXF157**).

Каждый школьник-линуксоид знает, что MINIX с самого своего рождения представляла собой микроядерную операционку. А вот какая она, эта микроядерность?

Для начала зададимся вопросом: а что же такое ядро вообще? Традиционно ядро определяется как программа, обеспечивающая взаимодействие всего остального системного и прикладного софта с аппаратной частью компьютера, а также распределение его ресурсов между приложениями. В соответствии с этим ядро функционирует в отдельной области памяти, которая так и называется – пространством ядра. Память же, отводимая под все остальные программы, именуется пользовательским пространством; протекающим там процессам доступ в пространство ядра закрыт, и как-либо влиять на ядро, в том числе и негативно (вследствие ошибок в программе или злого умысла), они не могут. Но все процессы внутри пространства ядра взаимодействуют друг с другом, и ошибка в одном из компонентов может повлечь за собой тяжкие последствия, вплоть до краха системы.

Как повысить надежность



Один из «косметических» вариантов решения проблемы был предложен Мэттом Диллоном [Matt Dillon] и реализован им в системе DragonFlyBSD, о которой рассказывалось в прошлой заметке цикла

(см. **LXF157**, май 2012 г.). Прошедшие с ее появления годы доказали жизнеспособность этой реализации. Однако проблемы «большого» монолитного ядра это не снимало.

Понятие аппаратуры компьютера, однако, оказывается двойным. С одной стороны, это те узлы, без которых машина в принципе не может функционировать, а именно – процессор и память. Эта сторона взаимодействия охватывается такими службами ядра, как обработчик прерываний, средства запуска и останова процессов, планировщик задач, механизм межпроцессного взаимодействия.

С другой стороны, в понятие аппаратной части включают и внешние, по отношению к системе процессор/память, устройства – от видеокарт и жестких дисков до принтеров, сканеров, сетевых адаптеров и многого, многого другого.

Более того, у этой аппаратной медали есть еще и третья сторона – сервисы доступа к файловым системам, сетевым протоколам и т.д. Они представляют собой связующее звено между собственно внешними устройствами и пользовательскими программами. Например, сервисы доступа к файловым системам обеспечивают возможность взаимодействия между дисковыми устройствами, несущими файловые системы, и обращающимися к ним приложениями.

Ядра, обеспечивающие все три рассмотренные функции, именуются монолитными. И они вполне успешно функционировали до тех пор, пока внешних устройств и сервисов было мало. Однако со временем количество и тех, и других стало разрастаться, как снежный ком. Вспомним, сколько на наших глазах появилось только критически важных устройств, таких, как новые дисковые интерфейсы или файловые системы.

В результате ядра стали катастрофически увеличиваться в размерах. Что влекло за собой а) непроизводительные расходы ресурсов, в первую очередь памяти, и б) рост числа ошибок, обусловленный грандиозными объемами «ядренного» кода, недоступного восприятию человека.

С первой болезнью научились бороться посредством модульного подхода: многие части ядра, обеспечивающие работу отдельных внешних устройств (т.н. драйверы устройств) или ядерных сервисов (в Linux их подчас также называют драйверами: например, «драйвер файловой системы имя рек»), могут не встраиваться жестко в исполняемый файл – образ ядра, а подключаться к нему в виде внешних модулей. И ныне ядра всех активно развивающихся UNIX-подобных систем, таких, как Linux или BSD, являются модульно-монолитными (подчас их называют и просто модульными).

Внешние модули могут как загружаться в память при старте системы, так и подгружаться в работающей системе, по необходимости. Нередко они обладают и способностью выгружаться, когда эта необходимость пропадет. Однако в любом случае модули функционируют в пространстве ядра, так что их введение проблемы общей устойчивости не решает: криво написанный драйвер устройства все равно сохраняет способность обрушить систему. А поскольку, как уже говорилось, сложность кода ядра растет, возрастает и вероятность появления ошибок, в том числе и критических для работы системы.

Для кардинального решения проблемы устойчивости ядра и были придуманы микроядра. Идея их в том, чтобы, по словам Таненбаума, «вынести ядро за пределы ядра». То есть оставить в ядре только средства управления базовой аппаратурой, а драйверы устройств и сервисы выделить в отдельные программы, функционирующие в пользовательском пространстве памяти. При необходимости они обращаются к функциям ядра через специальные процедуры, но влиять на него каким-либо образом не могут. Такой подход приводит к повышению надежности системы, но снижает производительность, поскольку требует дополнительных накладных расходов.



► MINIX 3 избрал своим тотемом забавного енота.

Собственно, идея микроядра появилась очень давно, чуть ли не одновременно с самим UNIX. Однако долгое время производительность машин не позволяла эффективно использовать микроядра в составе практически применяемых систем. Тем не менее, различных их реализаций было создано очень много. Время от времени то или иное микроядро пропагандировалось как основа операционной системы будущего, но удачных реализаций цельных микроядерных систем оказалось довольно мало.

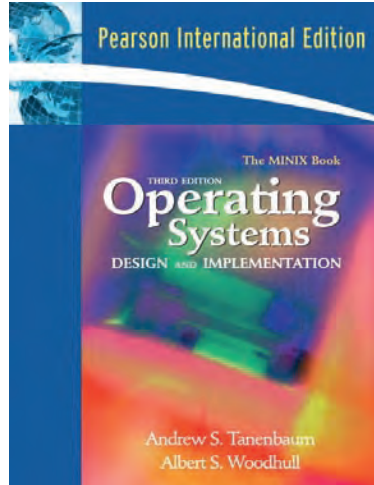
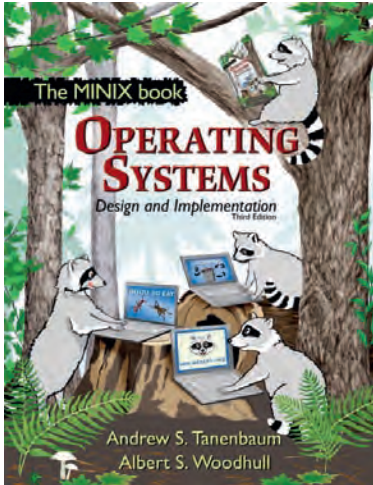
Среди удачных решений на базе микроядра наибольшей известностью пользуется QNX. Правда, эта система лишь недавно начала приоткрывать свой код, и потому об ее устройстве известно немного. Да и как ОС общего назначения она никогда не позиционировалась.

Из свободных микроядерных реализаций наибольшую известность приобрело микроядро Mach, разработывавшееся вплоть до второй половины 1990-х годов университетами –

сначала Карнеги–Меллона [Carnegie Mellon University], а затем штата Юта [University of Utah]. **Различные его версии в разное время составляли основу законченных систем, как свободных – BSD Mach или Yamitt, так и проприетарных – NeXT и продолжившей ее дело MacOS X.** Все они имели в своем составе микроядро Mach, поверх которого запускалась драйверно-сервисная часть от ядра BSD, собранная в виде отдельного модуля. Впрочем, из всех перечисленных систем только BSD Mach можно назвать по настоящему микроядерной, так как у него BSD-окружение ядра функционировало в пользовательском пространстве. И у NeXT, и у MacOS X BSD-окружение запускалось в том же пространстве ядра, так что микроядерными их можно назвать только по имени. А Yamitt в том виде, в каком я ее наблюдал, просто не запускалась вообще.

► «Отцы-основатели» — Эндрю Таненбаум (слева) и Брюс Званс.





► Учебник профессора Таненбаума выдержал не одно переиздание и стал классическим.

Наконец, как мы видели, начиная с 1987 года и по настоящее время существует MINIX 1/2, основанная на собственной реализации микроядра. Каковую, в рамках очерченных для нее задач, можно считать удачной. А вот будет ли сопутствовать удача ее потомку – микроядру нашей героини, MINIX 3, рассудит история.

Собственно о MINIX 3

Для начала заметим, что Таненбаум, по его же словам, не ставил самоцелью создание именно микроядерной ОС: задачей его команды было просто построение системы надежной и безопасной. Другое дело, что решение этой задачи им всегда виделось именно в микроядерной архитектуре. Почему он и предложил таковое, причем в виде, кардинально отличающемся от всех предлагавшихся ранее реализаций.

Отличие первое: микроядро MINIX 3 – самое микроядерное микроядро в мире. Из него вычищено все, кроме таких компонентов, как обработчик прерываний, средства запуска и останова процессов, планировщик задач и механизм межпроцессного взаимодействия; правда, почему-то в ядро включен и один из сервисов – сервис часов. В результате все это хозяйство укладывается менее чем в 4000 строк кода – и только оно подлежит исполнению в пространстве ядра.

Второе отличие – драйверная и сервисная части, вычлененные из ядра, разделены между собой. В результате образуется знаменитая четырехслойная модель-метафора, в основании которой лежит ядро, надстраиваемое «драйверным слоем», которое, в свою очередь, перекрывается «сервисным слоем» и венчается «слоем пользовательских программ».

Кроме того – и это третье отличие – каждый драйвер и каждый сервис представляет собой отдельный процесс в пользовательском пространстве, аналогично обычным пользовательским приложениям. В результате ни один драйвер и ни один сервис, как бы криво они ни были написаны, не в состоянии обрушить всю систему – точно также, как в любом UNIX'е это (почти) никогда не могут сделать обычные пользовательские приложения. Не повлиять им и на соседние процессы, так как напрямую взаимодействовать они не могут, а вынуждены при необходимости обращаться к ядру.

Наконец, четвертое, и, пожалуй, главное: сервер реинкарнаций. Это процесс, выступающий родительским по отношению к процессам всех драйверов и сервисов. Которые он запускает

при старте, а в дальнейшем отслеживает состояние. Если процесс какого-либо драйвера или сервиса в силу неких причин самопроизвольно «умирает», он запускает его вновь. Если один из драйверов или сервисов начинает вести себя «нехорошо», сервер реинкарнаций в состоянии убить соответствующий ему процесс и тут же запустить его заново, обеспечивая, таким образом прозрачное для пользователя самовосстановление системы при отказе почти любого драйвера устройства или системной службы.

Очевидно, что такая – достаточно сложная – схема взаимодействия драйверов и системных служб не может не привести к некоторой потере производительности по сравнению с обычными системами, где драйверы и сервисы сосуществуют в едином пространстве памяти, вне зависимости от того, пользовательском или «ядерном» (а подчас еще и вместе с ядром). То есть – неизбежно должны вызвать снижение быстродействия системы. В каких масштабах? Исследования команды Таненбаума дают ответ на этот вопрос, и к нему я еще вернусь. Но сначала сделаю маленькое отступление на тему, что такое быстродействие операционной системы вообще и с чем его едят, то есть – чем меряют.

Когда обсуждается проблема быстродействия любых ОС, первым делом обычно вспоминают о скорости загрузки. Вероятно, потому, что ее проще всего измерить: достаточно посидеть с секундомером перед несколькими одинаковыми машинами с разными операционками или дистрибутивами, чтобы потом уверенно утверждать об их сравнительном быстродействии.

Однако имеет ли отношение скорость загрузки системы к быстродействию ее при реальной работе? Отнюдь. Достаточно вспомнить, что MS DOS 3.3 на IBM PC/XT грузилась быстрее, чем любой Linux на любом супер-мега-Ivy Bridge. Перефразируя слова Сергея Образцова, измерение скорости загрузки ОС, строго говоря, даже к скорости загрузки ОС никакого отношения не имеет. Потому как зависит скорость загрузки в первую очередь от количества подгружаемых модулей и стартовых сервисов. Так что на самом деле мы определяем радиус кривизны рук пользователя, меру его лени или, напротив, количество свободного времени, которое он способен выделить на доведение системы до ума. А в последнее время,

с распространением SSD-накопителей, скорость загрузки ОС вообще измеряет толщину кошельки пользователя или степень его жадности.

Не лучше и с тестами на реальных приложениях под разными ОС.

Например, со столь любимым сравнением скорости обработки запросов web-сервером под Linux и FreeBSD, на основании чего делается вывод о превосходстве одной операционки над другой. Кстати, и не помню даже, кого над кем, да это и не важно. Потому что сразу же возникает закономерный вопрос: а что меряется в этом случае? Сравнительное быстродействие ОС? Или все-таки качество реализации конкретной версии *Apache* и, например, *MySQL* под ту и другую систему?

В общем, отдав в свое (не такое уж давнее) время дань увлечению всякого рода тестированием (это занятие было бы точнее классифицировать как пузомерию или... ну, как то, что в этнографической литературе называют «сравнением мужей»), я пришел к стойкому убеждению, что в большинстве случаев это либо измерение аршином с точностью до ангстрема, либо, по изящному выражению Таненбаума, сравнение яблок с апельсинами.

И тем не менее, методика тестирования, предложенная командой Таненбаума, производит впечатление. Во-первых, она (команда) поставила себе целью оценить влияние на быстродействие системы одного-единственного фактора: выноса драйверов за пределы ядра и перемещения их в пользовательское пространство. И потому тестирование быстродействия MINIX 3 про-

«Решение задачи выделось именно в микроядерной архитектуре.»

водилось... на MINIX 2. Каким образом? Очень просто: в качестве сравнительных объектов использовались каноническая MINIX 2, с одной стороны, и она же, пересобранная с удалением из ядра драйверов устройств и еще некоторыми модификациями, что фактически превратило ее в MINIX 3.

Во-вторых, в качестве тестов выполнялись процедуры, скорость которых действительно зависит от ОС исключительно или очень существенно: время исполнения системных вызовов, скорость чтения из файла и записи в файл, а также чтения непосредственно из блочного устройства (винчестера). Тесты с реальными приложениями тоже проводились – но предельно простые (в смысле, мало подверженные посторонним по отношению к ОС влияниям): пересборка образа системы и набора контрольных тестов POSIX, а также обработка текстового файла утилитами типа *sed*, *grep*.

Результаты оказались парадоксальными. Разумеется, квази-MINIX 3 проиграла MINIX 2 по всем статьям. Но давайте посмотрим, где и насколько. По чисто «ядерным» тестам исполнения системных вызовов отставание первой составило 12 %, по тестам на файловых операциях и запросах к базам данных – 8–9 %, по тестам на реальных приложениях – 6 %. Последнее на современных машинах практически равнозначно отсутствию визуальной разницы вообще.

На пути к возможностям

Результаты тестов, о которых я только что говорил, были опубликованы в работе “A Lightweight Method for Building Reliable Operating Systems Despite Unreliable Device Drivers” еще в январе 2006 года (PDF-версию можно найти на <http://www.minix3.org/doc/reliable-os.pdf>). Они показали, что катастрофического провала производительности при переходе к «чистому» микроядру не наблюдается. И, следовательно, идея, положенная в основу архитектуры MINIX 3:

- » имеет право на существование, и
- » заслуживает дальнейшего развития.

Так что дело оставалось за малым: воплотить идею в нечто реально работающее. А до этого на момент публикации результатов тестов MINIX 3 было так же далеко, как омарам – нетрадиционным способом до города Пекина.

Обычно, когда описывают особенности некоей системы, разговор начинается с того, что в ней есть. Я же нарушу традицию, и расскажу о том, чего в MINIX 3 не было. И даже не в момент ее анонса, а более чем год спустя, на рубеже 2006/2007 годов, когда мне довелось познакомиться с ней воочию – в актуальной на тот момент версии 3.1.2.

Итак, в MINIX 3 отсутствовали:

- » поддержка огромного количества современного оборудования – от шины USB до интерфейса SATA, а видеоподсистема обеспечивала работу только в VESA-режиме;
- » возможность динамической линковки приложений с функциями системных библиотек;
- » поддержка каких-либо файловых систем, кроме своей собственной – даже доступ к ISO 9660 осуществлялся через устройство, которое у людей располагается обычно чуть ниже спины;
- » поддержка виртуальной памяти.

Ясное дело, что с прикладным софтом дело обстоит не лучше. В свежее установленной системе имелся набор классических UNIX-утилит в реализации, примерно соответствующей стандарту POSIX, то есть далеко не самых богатых возможностями. Конечно, эту проблему можно было частично решить путем доустановки дополнительных пакетов (а их уже тогда хватало). Но вот с поддержкой устройств, файловых систем и прочего системного инвентаря рядовой пользователь ничего поделать не мог – это была вахта разработчиков.

Чего нет у рыб?

По поводу возможностей, отсутствующих в MINIX 3 на момент моего с ней знакомства, хочется сделать небольшое литературное отступление.

Выдающийся русский филолог и писатель Лев Успенский (автор, в числе многого прочего, замечательной книжки «Слово о словах») в своих воспоминаниях рассказывает, как в студенческие годы сдавал экзамен по какой-то биологической дисциплине. И на вопрос, вынесенный в заглавие этой врезки, ответил: «У рыб нет

монокля и полного собрания сочинений Шпильгагена». За что и удостоился отличной отметки.

С MINIX 3 – случай аналогичный. Монокль к ее установочному диску не прилагается, и полное собрание сочинений Шпильгагена на нем не присутствует (да и вряд ли вообще существует в оцифрованном виде). Однако, как и с рыбами, список отсутствующих в MINIX 3 функций моноклем и Шпильгагеном далеко не исчерпывается.

И они стоят на ней доблестно: постепенно в MINIX 3 появилась поддержка виртуальной и разделяемой памяти, иных файловых систем, вплоть до подсистемы FUSE с экспериментальной поддержкой NTFS. Обрастала она и драйверами устройств, расширялся круг портируемых приложений, в том числе за счет задействия системы *pkgsrc* – той же самой, что была принята на вооружение в DragonFlyBSD.

Описывать в деталях хронологию всех этих изменений я не буду: заинтересовавшийся читатель легко может отследить их по новостному разделу официального сайта или русскоязычного его гомолога. На последнем, кроме того, можно видеть, как крепла поддержка в MINIX 3 русского языка – сначала усилиями отечественных добровольцев, а потом уже и на официальном уровне.


И ныне, пожалуй, в MINIX 3 не найти разве что, действительно, монокля и полного собрания сочинений Шпильгагена – большинство остальных атрибутов полноценной операционки в ней имеется. Что внушает оптимизм относительно его дальнейшего будущего.

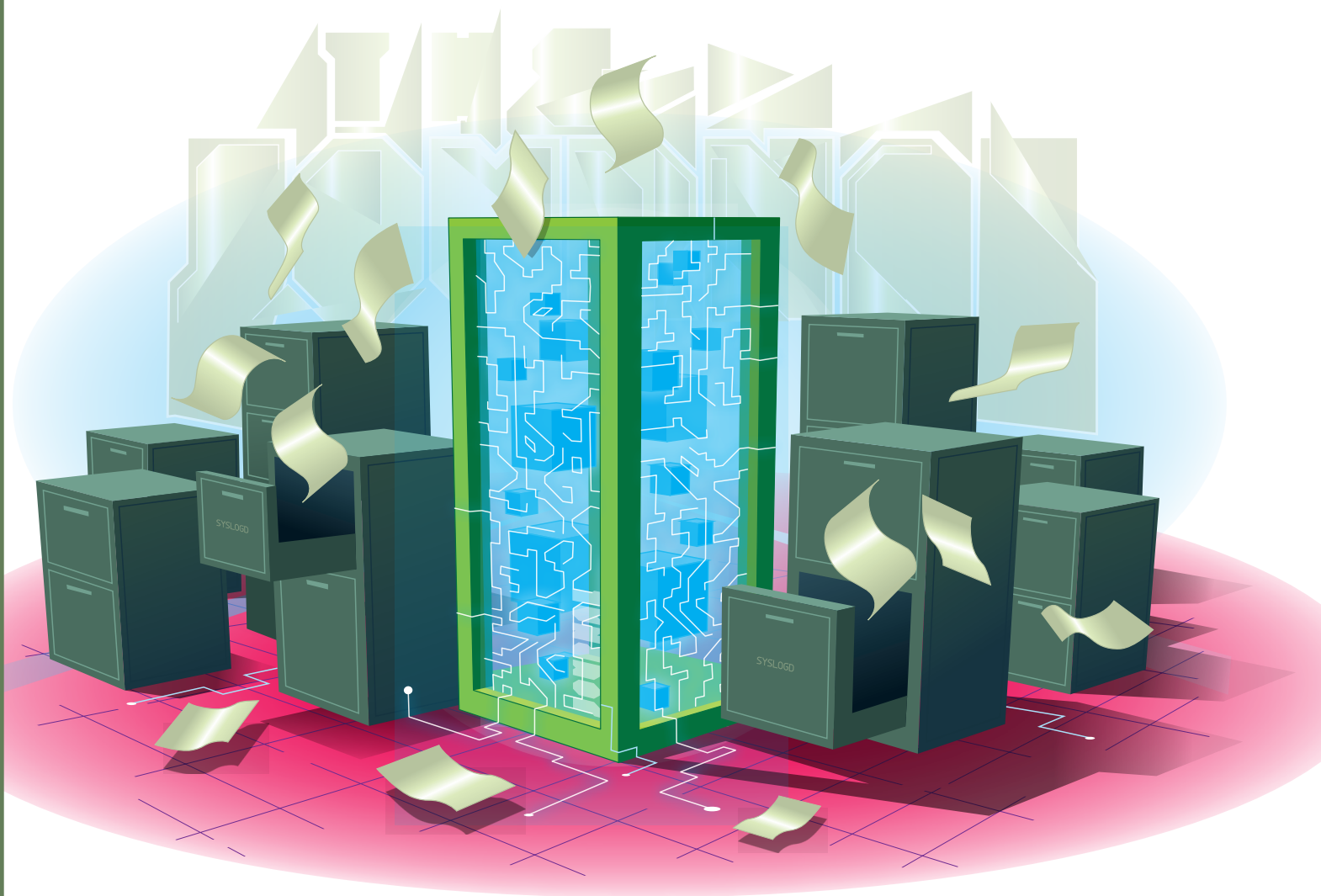
Заключение

Вот о будущем всех трех родившихся на наших глазах операционных систем я и хотел бы сказать несколько слов под занавес. С одной стороны, и DragonFlyBSD, и MINIX 3 развиваются – может быть, не такими темпами, как хотелось бы их поклонникам, но поступательно и необратимо. Правда, о Syllable этого не скажешь – ее жизнь протекает ни шатко, ни валко. Но относительный успех хотя бы двух систем из трех, на фоне бурного развития Linux'a, показателен. Это при том, что разработчики Free- и других BSD тоже не сидят сложа руки.

С другой стороны, Год Великого перелома, о котором речь пойдет в одной из следующих статей цикла, смешал карты: сначала интенсивная десктопизация Linux, а затем внедрение ее инкарнации – Android'a, казалось бы, не оставил для всех других операционных систем места под солнцем.

Однако у этой медали, как ни странно, есть и третья сторона: Linux буквально в последнее время пошел по пути, от которого в восторге далеко не все пользователи, майнтайнеры и разработчики. И кто знает, не сделает ли это востребованными «более иные» операционки – как давно существующие, так и недавно рожденные.

Однако тут мы переходим в область, где кончается история и начинается политика, да еще и приправленная предсказаниями и пророчествами. И прежде чем ступить на эту зыбкую почву, я хотел бы завершить цикл рассказов о прошлом – давнем и недавнем. А там – чем черт ни шутит. Может быть, страсти сегодняшнего дня к тому времени тоже превратятся в достояние истории... 



Что за штука...

The Journal?

Джонатан Робертс изучает новый инструмент для записи системных логов.

В Разве мы не говорили о журналах совсем недавно?

О Верно, говорили, в связи с *Zeitgeist*. Но это была настольная программа для записи ваших посещений файлов, чтобы потом легче найти эти файлы.

В А чем этот новый *Journal* отличается от старого?

О *The Journal* – это системный журнал. Он не ищет файлы, а записывает информацию о работе системы: данные об использовании

ресурсов, сведения о безопасности и ошибках – все это хранится в системных журналах.

В Но Linux же и так записывает эту информацию. Затем и нужны все эти *.log* файлы в */var/log/messages*, верно?

О Да, это так. Когда системным процессам есть что записать, они в фоновом режиме посылают сообщение в соответствующую программу. Происходит это по протоколу, известному как *syslog*, а само приложение называется *syslog daemon*.

В Да, точно, многие фоновые процессы называются демонами. Так этот *syslog daemon* пишет в журналы?

О Пишет, в некоторые. У других приложений бывают и собственные системы журналирования, и собственные форматы файлов. Логи, составленные *syslog*, обычно пишутся в форме обычных человеко-читаемых текстовых файлов.

В Хорошо, сколько бы их ни было, зачем вообще нужна новая система журналирования? Ведь Linux сейчас себя проявляет очень

хорошо, даже в сферах очень ответственных; ведь не может же нынешняя система логирования по умолчанию быть плохой?

О Она и не плоха, и действительно многим впечатляет, но ей, как-никак, 30 лет. За это время были выявлены уязвимости, несоответствия разрослись в отдельные решения, и появились новые программы, способные использовать журналы так, как не догадались бы прежние создатели *syslog*.

В Это все общие слова, а вы не могли бы привести реальный пример?

О Конечно. На самом деле, разработчики *The Journal* сумели привести порядка 14 таких примеров.

В Аж целых 14! Вы вроде говорили, что *syslog* был не так уж плох.

О Вот именно, и разработчики *syslog* всерьез усомнились в том, что все 14 реально правомерны. Но в итоге с несколькими они согласились. В том числе – с проблемой отсутствия правил, определяющих формат записываемого лог-файла, и проблемой индексации, возникающей из-за того, что эти файлы неотличимы от обычных текстовых.

В А что тут плохого, ведь в Linux столько приложений, что не обойтись без разных требований к журналированию. Использование одного формата может ограничить свободу их работы.

О Действительно, малое количество требований к формату сообщения делает систему более гибкой и мощной, но в то же время усложняет чтение журналов. Пусть вам нужно написать программу для просмотра журналов ядра, X-сервера и почты: придется иметь дело с тремя разными форматами. Это отнимет время и увеличит объем кода, а вместе с ним и вероятность ошибок.

В Да, это и правда проблема. Но я не вполне убежден, что текстовый формат неудобен – ведь его можно читать при помощи простого текстового редактора, и вы никогда не забудете, как это делается. Это в стиле Unix!

О И да, и нет. Открытый текст действительно обладает названными вами преимуществами, но такие файлы нельзя проиндексировать, и их долго искать. Например, чтобы найти в журнале все HTTP-запросы с конкретного IP-адреса, придется использовать линейный поиск, а в вычислительной среде – это очень медленно.

В Линейный поиск? А вы не поясните, что это такое?

О Проще всего представить себе новогоднюю гирлянду, в которой не работает всего одна лампочка, но из-за нее не горит и вся гирлянда

в целом. Чтобы найти неисправную, вам придется перепробовать все по отдельности! А теперь прикиньте, что значит проделать такое с десятками тысяч единиц данных.

В Так, суть проблемы ясна. Но, подозреваю, *The Journal* с этим справится?

О Именно. Первая проблема снимается за счет унификации формата журналов. Каждая запись состоит из ряда полей ключ/значение. Поэтому каждая программа, обращаясь к журналам, будет искать нужный ключ в любой записи, таким образом находя искомое значение.

В То есть формат файла работает по принципу словаря? Ключ – это нужное слово, а значение – это определение?

О Да, так и есть. Сравните это с текстовыми файлами *syslog*, где тип данных, сами данные и другая информация перемешаны в кучу – причем в каждом файле по-своему!

А такой подход не только облегчает доступ к файлам, но и достаточно гибок. Запись может состоять из множества разных полей, но они всегда будут записаны интуитивно понятным и доступным способом.

В Хм... интересно. Но как это связано со скоростью нахождения информации?

О Напрямую – никак. Но здесь поможет еще одна важная функция *The Journal*. Каждому лог-файлу теперь присваивается собственный длинный номер, называемый UUID. Он будет уникальным не для каждого файла в отдельности, а для конкретного типа журналов. А значит,

рует эти службы, заставляя их запускаться быстрее и давая возможность перезапуска и остановки.

О Браво, приятно видеть, что собеседник тебя действительно слушает. Поскольку *systemd* управляет службами, многие из которых являются программами, записывающими информацию, *systemd* много о них знает. А поскольку *The Journal* находится в составе *systemd*, он также может пользоваться этой информацией. К примеру, он может собирать и, что еще важнее, проверять подлинность массы контекстной информации, полезной для отладки программ: ID процессов, места хранения исполняемых файлов, и т.д.

В Проверять подлинность? Похоже, это насчет безопасности.

О Да. Ведь по сути, любая служба может отправить сообщение в *syslog* и записать туда любую информацию, а *syslog* никак не проверяет ее истинность. А злонамеренные программы могут воспользоваться этим для эмуляции законных программ и маскировки своих действий в журнале, чтобы вредить незаметно. Есть и много других функций. Читайте о них в подробном анонсе выхода *The Journal* (<http://tinyurl.com/853f73n>).

В Непременно. Еще вопрос: каковы у *The Journal* шансы на выживаемость?

Никто не любит перемен, особенно если есть действующая альтернатива.

О Очень хороший вопрос. Поскольку *The Journal* интегрирован в *systemd*, уже входящего в состав многих дистрибутивов, есть все шансы на то, что он обретет популярность. Тем не менее, су-

«The Journal — это компонент systemd, новый диспетчер служб и система загрузки.»

можно создать инструментарий для распознавания типа журнала, чтобы затем просматривать поля только в них и находить необходимую информацию.

В Многообещающее начало, но для полностью новой системы новых функций как-то маловато.

О На самом деле, их множество. Из того, что мы еще не упомянули, *The Journal* ведь является компонентом *systemd* – это новейший диспетчер служб и система загрузки, используемая многими дистрибутивами Linux.

В О да, об этом вы уже говорили. При включении компьютера стартуют разнообразные «процессы» – фоновые программы, поддерживающие работу компьютера. *Systemd* контроли-

рует некое предубеждение против самой идеи *The Journal*. Одни сомневаются в его безопасности, другим жалко расставаться с текстовыми файлами, третьи считают, что лучше было бы подправить *syslog*, чем изобретать велосипед. Не говоря уж о том, что разработчикам многих других проектов придется переписать свои программы, чтобы воспользоваться новыми функциями *The Journal*.

В Так значит, следует ожидать грядущих боев?

О Можно и так сказать. Но не забудем о том, что на счету разработчиков *The Journal* как минимум еще несколько весьма противоречивых решений для Linux, включая небезызвестный *PulseAudio*. И тем не менее все они вошли в состав большинства дистрибутивов, так что, похоже, эти ребята знают, что делают. **LXF**



По рецептам доктора Брауна

Д-р Крис Браун

Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Назвался груздем...

Моя жена ведет кружок по пошиву лоскутных покрывал. Поотиравшись около их сборищ, я научился отличать узор «плетенка» от узора «лестничка», узнал, что дешевый ватин мохрится, и понял, как вырезать фестоны для «широких квадратов». Нет, я не выдумывал эти термины.

Свой жаргон есть у всех профессий. Но я бы сказал, что в мире открытого ПО его многовато. Где еще вы услышите, что обратный слэш экранирует звездочку, или получите советы погрелить в логах, отправить демону сигнал HUP, су'кнуть в root, пропатчить ядро или загрузить Grub? Где вы еще сможете запереть руткит в песочнице или уловить вредоносное ПО на «медовую приманку» [honeypot, ресурс-приманка, вызывает атаку на себя для изучения стратегии злоумышленника и определения защиты, – прим. пер.]? Где еще вы узнаете, что парень, с которым вы только что взяли по пиву, все еще работает в Lucid с «предками пространства ARM»? Эти термины я тоже не выдумал.

...надевай шляпу

Все это (в чем-то) подводит меня к недавно попавшейся мне книге «*Gray Hat Hacking* [Учебник серого хакера]», примечательной потрясающим набором охваченных тем.

Вы научитесь вводить в заблуждение охранные программы и создавать «медовую приманку». А зная Ассемблер, вы сможете подробно разобрать создание эксплойта и проведение анализа на уязвимости. Этически такие книги балансируют на грани между обучением «черных шляп» (плохих парней) и «белых шляп» (хороших парней).

В первых главах внимание уделяется вопросам этики, с основной идеей «нельзя должным образом защитить себя от угроз, которых не понимаешь». Эта книга не для слабаков (во многих аспектах), но если в названии вашей должности есть слово «безопасность», ее стоит прочесть.

chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



Знать свои слабости

Чтобы защитить свои компьютеры от нападения, нужно знать свои уязвимости. Вам поможет *OpenVAS*.

Можно ли быть уверенным в том, что система защищена? Ну, на 100 % нельзя никогда. Но можно существенно повысить свою уверенность, запустив сканер уязвимостей, и во многих организациях это делается на регулярной основе.

OpenVAS – одна из таких программ. Это отвлечение *Nessus*, популярного сканера, который в 2005 году превратился из открытого в коммерческий. К моменту чтения этой статьи должна выйти *OpenVAS 5*. *OpenVAS* запускает на главном компьютере сканирующий демон, который выполняет серию проверок на уязвимость для заданного набора объектов. Эти проверки называются проверками сетевой уязвимости (*NVT – network vulnerability tests*) и пишутся на простом скриптовом языке (*Nessus Attack Scripting Language – язык описания атак Nessus*).

Тысячное войско

Сейчас таких тестов чуть менее 25000, и каждый день добавляются новые в ответ на новую информацию об угрозах безопасности, поступающую от производителей ПО и из отчетов исследователей. Около 16000 из этих тестов основаны на информации от производителей ПО, обычно в такой форме: «О, у вас версия X приложения Y. Значит, у вас есть уязвимость Z». Другие тесты реально взаимодействуют с системой, определяя уязвимости. Качество работы *OpenVAS*, очевидно, зависит от качества тестов и своевременности их создания и распространения по мере обнаружения новых угроз. *OpenVAS* обеспечивает бесплатный приток

тестов; есть также и коммерческий, от немецкой компании Greenbone, разработчика *OpenVAS*. Чтобы по-быстрому это попробовать, скачайте виртуальный экземпляр *OpenVAS4* с сайта openvas.org; он совместим с *VirtualBox* и *VMWare* (сам я пробовал только последнее) и предоставляет сервер *OpenVAS* – с ним можно работать через утилиты командной строки или с помощью браузера через порт 9392. Надо сказать, что хотя его легко загрузить и установить, кривая его изучения довольно крута. Для практического использования установите сами пакеты *OpenVAS*. Возможно, они есть в репозиториях вашего дистрибутива.

Наконец, заострим, что запуск сканера уязвимостей сам по себе волшебным образом не повысит вам безопасность. Кто-то должен его отчеты читать и принимать меры. И во многих случаях достаточно просто обновить уязвимое приложение до более поздней версии.

Взгляд изнутри

Существует много других утилит сканирования. *Bastille* – утилита сканирования и укрепления системы, которая взаправду усилит безопасность системы и предложит краткий курс по безопасности в качестве бонуса. Проект некоторое время дремал, но, кажется, возвращается к жизни (www.bastille-unix.org). В отличие от *OpenVAS*, *Bastille* проверяет систему «изнутри», т.е. вы запускаете его на сканируемом компьютере.

Нордавинд Северо-Запад

Охранные системы на базе GNU/Linux

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

Наши услуги и продукты

- Проведение НИР
- Разработка и изготовление систем охранного телевидения и сопутствующих компонентов
- Проектирование сложных систем и комплексов
- Широкий ассортимент готовой продукции

Наши системы видеонаблюдения предназначены для:

Паркингов

Зданий

Бизнес-центров

Торговых комплексов и складов

Открытых площадок

Жилых домов

Объектов повышенной секретности

Режимных предприятий

Сертификаты ФСТЭК и ФСБ

4 канальная система видеонаблюдения стоимостью **40 тыс. руб.** —
в 2 раза дешевле, чем предыдущие технологии.



ТелеВизард

Многофункциональный сервер общего назначения.



ТелеВизард Авто

Самая бюджетная и высокоэффективная система распознавания автомобильных номеров.



Источники видеосигнала

Комплекс eyeSense, в состав которого входят видеочамера и объектив. Дополнительное оборудование — ИК-прожектор и климат-защита.

«Нордавинд Северо-Запад»

Научно-производственная компания

Тел. +7 (812)

309 0686

Тел./факс +7 (812)

640 4990

<http://spb.nordavind.ru>

e-mail: spb@nordavind.ru

Приглашаем к партнерству дилеров и проектировщиков систем

Linux упрочняет защиту

У SELinux репутация невероятно сложной системы, но он слишком важен, чтобы его игнорировать. Рассмотрим этот важный рубеж обороны.

Должен признаться, что при упоминании SELinux я веду себя как страус – прячу голову в песок и надеюсь, что он исчезнет. Но RedHat принимает его всерьез, и системным администраторам приходится иметь с ним дело все чаще. И я решил, что лучше рассказать об его основах, хотя бы немного.

Прежде всего, зачем он нужен? В двух словах – его назначение в том, чтобы ограничить ущерб системы из-за атаки, воспользовавшейся уязвимостью в программе. Чтобы понять, как это работает, начнем с традиционной модели прав доступа к файлам в Linux, под которой я имею в виду известные разрешения **rwX**. В этой модели авторизация происходит исключительно по UID и GID процесса, выполняющего запрос. Если процесс запускается от имени пользователя **chris**, я вижу одни права доступа, если от имени **root** – другие, и т. д. Например, пусть я пользуюсь *Firefox*. Разумеется, браузеру нужен доступ к некоторым каталогам файловой системы – чтобы кэшировать cookie, записывать историю и т. д., но все мои файлы ему ни к чему. Однако если открыть в нем активное вредоносное содержимое, умеющее использовать уязвимость в браузере, злоумышленник потенциально получит доступ к тем же файлам, что и я (как пользователь **chris**).

С сетевыми сервисами проблема часто и того острее, ведь многие из них запускаются от имени суперпользователя-**root**, и воспользовавшись уязвимостью в таких сервисах, атакующий может получить доступ ко всей системе.

Как написано в руководстве пользователя по SELinux в RedHat, «Многие системные сервисы и программы запускаются со вчерне заданными привилегиями, намного превосходящими их потребности, и ошибка в любой из этих программ может быть использована для получения доступа к системе».

Здесь традиционная модель безопасности в Linux становится бессильной, и управление на себя берет SELinux. Важно понимать, что решения об авторизации, принимаемые SELinux, дополняют, а не заменяют традиционную модель. Если права **rwX** не разрешают доступа – точка. Но если они разрешают, политики SELinux дают системе дополнительную возможность запретить его.

Уязвимости дня 0

Уязвимость программы – недостаток программы, позволяющий злоумышленнику заставить программу сделать нечто для нее нештатное. В идеальном мире у программ не было бы уязвимостей (вообще-то в истинно идеальном мире не было бы и злоумышленников, но это другая история).

В реальности уязвимости иногда обнаруживаются. И тогда разработчики программ должны срочно взяться за проблему и выпустить заплатку или обновление, а системные администраторы (это вы!) должны установить его.

Но как бы быстро разработчик ни создал заплатку и как бы быстро администратор ни установил бы ее, существует промежуток времени, в который уязвимость известна, но не исправлена. Это так называемые уязвимости нулевого дня, известные потенциальным злоумышленникам, но неизвестные разработчикам ПО.

Они-то и вызывают у системных администраторов приступы паранойи, и служат двигателем разработки систем управления доступом вроде SELinux.

Невообразимая сложность

Детали устройства SELinux, мягко говоря, непросты. Но давайте окунемся в них и узнаем, сможем ли мы хотя бы почувствовать их аромат. Если начинать с самого начала, то у каждого файла и каждого процесса в системе SELinux есть контекст безопасности, в который входят четыре составляющих:

ПОЛЬЗОВАТЕЛЬ [USER]	Это пользователь SELinux, которому «принадлежит» файл или процесс. Эти пользователи напрямую не соответствуют учетным записям в <i>/etc/passwd</i> ; на самом деле, большинство обычных учетных записей в Linux связаны с пользователем unconfined_u SELinux.
РОЛЬ [ROLE]	Используется для управления доступом на основе ролей (role-based access control – RBAC). Пользователи авторизуются для получения определенных ролей, а роли определяют, к каким ресурсам у пользователя есть доступ. Роли используются в управлении процессами, но файлы, которые по сути пассивны, на самом деле не используются ролями, и для них используется «пустая роль» object_r .
ТИП [TYPE]	Это главные атрибуты безопасности, которыми SELinux руководствуется при принятии решения о доступе. При применении к процессам типы также называются доменами и по сути определяют именованную «песочницу», в которой выполняется процесс.
УРОВЕНЬ [LEVEL]	С помощью этого атрибута реализуется многоуровневая система доступа – в ее основе лежит идея о том, что документы могут быть «публичными», «конфиденциальными», «совершенно секретными» и т. д.

Привязка контекстов безопасности к файлам называется «пометкой» файловой системы.

Контекст безопасности файла можно просмотреть, скомандовав **ls -Z**:

```
# ls -Z /usr/bin/ssh
-rwxr-xr-x. root root system_u:object_r:ssh_exec_t:s0 /usr/bin/ssh
```

Здесь мы видим, что у файла есть собственный тип **ssh_exec_t**.

Обычный файл пользователя в домашнем каталоге может выглядеть так:

```
$ ls -lZ sample
-rw-r--r--. chris chris unconfined_u:object_r:user_home_t:s0 sample
```

Обратите внимание, что в качестве владельца SELinux отображается **unconfined_u**, как мы упоминали ранее.

Контекст безопасности процесса можно просмотреть командой **ps -Z**:

```
# ps -eZ | grep sshd
system_u:system_r:sshd_t:s0-c0:c1023 1508 ? 00:00:00 sshd
```

Здесь мы опять же видим, что у демона **sshd** есть собственный домен (**sandbox**, песочница) по имени **sshd_t**.

Вся эта новая информация, связываемая с файлами и процессами, требует изменения ряда стандартных утилит, чтобы

К истокам

SELinux изначально разрабатывался в Агентстве национальной безопасности США (в общих чертах – аналог Центра правительственной связи Великобритании, см. <http://www.nsa.gov/research/selinux>). Большинство других утилит также были разработаны компанией Tresys (см. oss.tresys.com), ей принадлежит большой набор утилит (как консольных, так и графических) для анализа и разработки политик.

они «знали» о SELinux. Например, *ls*, *ps* и *id* расширяются с целью отображения контекста безопасности файлов, процессов и пользователей соответственно. Утилиты управления файлами, типа *cp* и *mv*, изменяются так, чтобы сохранять контекст безопасности для создаваемых ими файлов. Даже у *tar* появляется возможность сохранять контексты безопасности файлов в архив (и восстанавливать их). Кроме того, есть несколько утилит SELinux для управления контекстами безопасности – к ним относятся *restorecon*, *secon*, *chcon*, *findcon* и *setfiles*. Быстрый поиск с помощью *apropos* нашел в общей сложности не менее 41 команды, имеющей отношение к SELinux.

Не наша это политика...

Все это вместе связывает SELinux Policy [политика SELinux], которая по сути представляет собой набор правил вида «процесс, выполняющийся в домене X, может получить доступ к файлам типа Y». Если вам нравится метафора с песочницей, то политики определяют границы песочницы. Вся система чрезвычайно детализирована. Чтобы дать вам представление о ее масштабе, скажу, что в пакете RHEL6 *selinux-policy* («образец политики») почти 400 файлов политик, а число строк в них превышает 350 000. Политика для одного только *Apache* состоит из 1400 строк. Впрочем, знайте, что перед загрузкой в ядро эти политики компилируются в двоичную форму.

Планируя эту статью, я наивно полагал, что сумею объяснить, как пишется политика. Но по мере моего продвижения стало ясно, что если наш главред не отдаст мне львиную долю этого номера, ничего не получится. Даже 82-страничное руководство пользователя по SELinux в RedHat и близко не подходит к написанию политик. Там даже не приведено примера. Существуют программы, которые помогают в анализе и разработке политик (например, *apol* и *SLIDE*, обе от Tresys), но это уже выходит за рамки данной статьи. Большинство администраторов будут расценивать политики SELinux так же, как исходные коды ядра – теоретически их при необходимости можно изменить, но на практике вы вряд ли будете этим заниматься. (Вы знаете разницу между теорией и практикой? Так вот, в теории нет разницы, а на практике она есть...)

Red Hat и Fedora (по-моему, единственные дистрибутивы, создающие что-то реально полезное с SELinux), проводят так называемую «целевую» политику. То есть, они в основном ограничивают сервисы, которые слушают сеть, а также несколько программ, которые изменяют *setuid* на *root*, таких как */usr/bin/passwd*. Они не ограничивают деятельность, скажем, вошедшего в систему пользователя. Так, например, контекст безопасности обычного пользователя выглядит так:

```
$ id -Z
```

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Прежде чем паниковать при мысли о пользователях с «неограниченными [unconfined]» правами, вспомните, что традиционную модель прав доступа в Linux никто не отменял, просто SELinux пока не настроен на дальнейшее ограничение доступа.

Настройка

SELinux позволяет в некоторой степени изменить настройки без необходимости переписывать политики. На самом общем уровне можно изменять значение параметра SELINUX в */etc/selinux/config* на Disabled [Отключен], Permissive [Толерантный] или Enforcing [Усиленный]. Назначение первого и третьего режимов понятно по названиям, а второй, пожалуй, требует пояснения. В этом режиме доступ, запрещаемый политикой, разрешается, но соответствующие действия записываются в файл журнала. С помощью этого режима удобно убедиться, что ваши настройки не выведут систему из строя после переключения в режим Усиленный. Есть даже утилита *audit2allow*, которая сгенерирует правила политики SELinux по лог-файлам операций, которым было отказано в доступе.

Определить, в каком режиме вы находитесь сейчас, можно командой *sestatus*:

```
# sestatus
SELinux status: enabled
SELinuxfs mount: /selinux
Current mode: enforcing
Mode from config file: enforcing
Policy version: 24
Policy from config file: targeted
```

Более тонкий контроль обеспечивается логическими правилами SELinux. С их помощью администратор может разрешить или запретить определенный доступ локально без необходимости перезагрузки или перекомпиляции политики SELinux. Примеры логических правил таковы: «Разрешить https доступ к файловым системам nfs», «Разрешить системе запускаться с NIS», «Разрешить samba открывать домашние каталоги пользователей для общего доступа». Команда

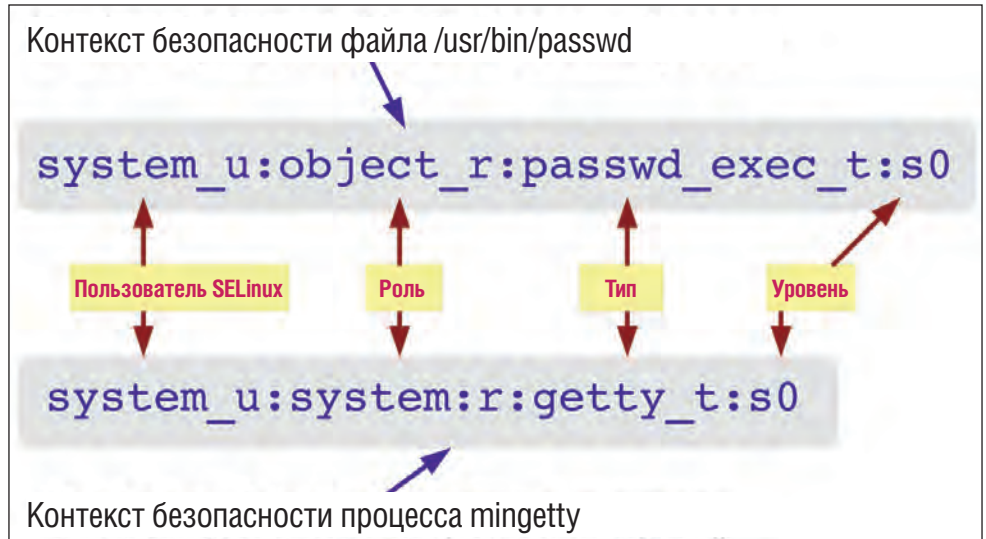
```
getsebool -a
```

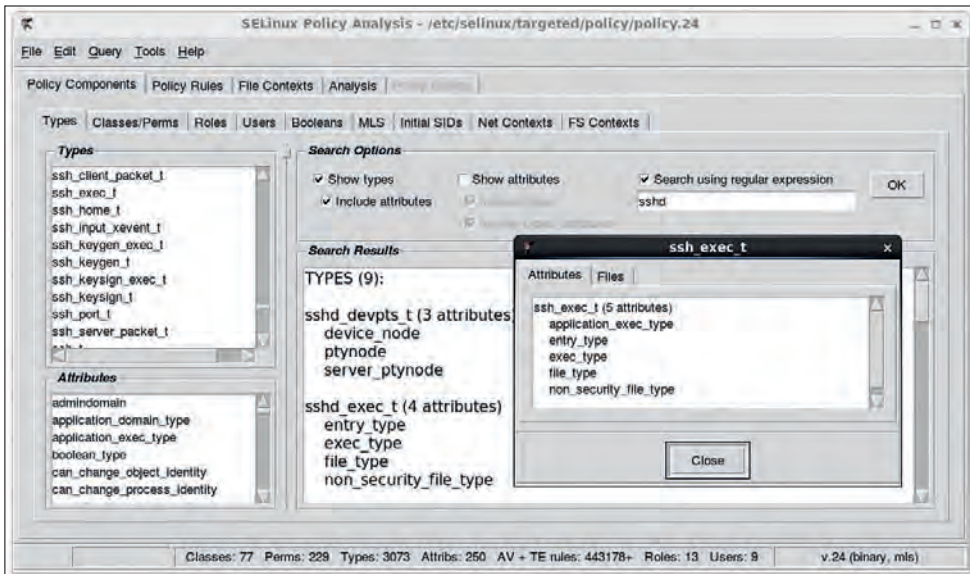
выведет все логические правила. В моей собственной тестовой системе RHEL6 их 168.

Бурный инцидент

Если вы играете не по правилам, SELinux запросто подпрыгнет и укусит вас. В качестве такого примера расскажу историю, имевшую место на одном из недавних занятий. Курс содержал

► Контексты безопасности, которые есть у каждого файла и каждого процесса – сердце SELinux.





passwd, которая сохранит верный контекст безопасности SELinux. Но мы сделали такое, от чего не знакомый с SELinux администратор не ждал бы ничего плохого, вот и огребли.

Проанализируем произошедшее подробнее. У исходного файла **/etc/shadow** была такая метка:

```
# ls -lZ /etc/shadow
----- root root
system_u:object_r:shadow_t:s0 /etc/shadow
```

После запуска скрипта метка стала такой:

```
# ls -lZ /etc/shadow
----- root root
unconfined_u:object_r:user_tmp_t:s0 /etc/shadow
```

Изменились и владелец файла в SELinux, и «тип» метки. После возвращения в нормальное состояние болтливая запись

► **Aprol**, одна из графических утилит от Tresys, помогает анализировать политики SELinux.

упражнение для студентов – восстановить утерянный пароль пользователя root. Чтобы «потерять» пароль, студенты запускали скрипт, который (тайно) модифицировал файл **/etc/shadow**, примерно такой:

```
sed s/foo/bar /etc/shadow > /tmp/newshadow
mv /tmp/newshadow /etc/shadow
chmod 400 /etc/shadow
```

Скрипт создавал новый, измененный файл с неработающим паролем root и переименовал его обратно в **/etc/shadow**. И студентам пришлось сделать спасательную перезагрузку системы и сбросить пароль root. В системе без SELinux у них бы все получилось.

Что же произошло? Им вообще не удалось войти в систему – ни от имени root, ни от имени любого другого пользователя. Проблема в том, что у файла **shadow** особый контекст безопасности SELinux. А у вновь созданной копии файла был другой контекст безопасности. Как следствие, после перезагрузки системы даже программа login (которая запускается от имени root) не смогла прочитать файл **shadow**. Результат? Входа нет! Хотя это прекрасно иллюстрирует мощь SELinux, именно в такие ситуации преподаватель меньше всего жаждет попасть на уроках; впрочем, у нас не было выбора.

Неусиленный режим

Нам удалось восстановить систему, прервав загрузку и добавив **enforcing=0** к списку параметров, которые **Grub** передает ядру. Тогда ядро запускается с неусиленным режимом SELinux, в котором мы по крайней мере смогли войти в систему. Чтобы полностью восстановить систему, мы выполнили команду:

```
# restorecon /etc/shadow
```

которая вернула прежний контекст безопасности файла.

В чем мораль сей басни? Конечно, не стоит менять файл **shadow** так, как это сделали мы. Нужно пользоваться командой

в **/var/log/audit/audit.log** сообщает нам, что произошло:

```
type=AVC msg=audit(11330372308.768:52621): avc: denied {
read } for pid=28670 comm="unix_chkpwd"
name="shadow" dev=dm-0 ino=815961
scontext=system_u:system_r:chkpwd_t:s0-s0:c0.c1023 tcont
xt=unconfined_u:object_r:user_tmp_t:s0 tclass=file
```

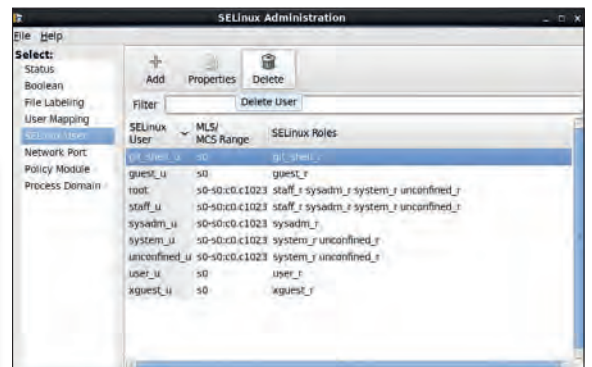
Взглянув на нее внимательно, вы увидите, что процессу в домене **chkpwd_t** был запрещен доступ к файлу типа **user_tmp_t**, а это тип нашего (испорченного) файла **shadow**. Тип **type=AVC**, который вы здесь видите – ссылка на кэш векторов доступа (Access Vector Cache), хранилище недавно принятых SELinux решений о разрешении или запрещении доступа.

Кстати, с помощью команды **aureport** можно извлечь сообщения SELinux из лог-файла в более удобном для восприятия формате.

И в заключение...

Возможности SELinux обширны. При верной настройке он может резко снизить риски взлома сервера, а при неверной – полностью вывести систему из строя. На мой взгляд, главная преграда к использованию SELinux – это его сложность. Системные администраторы любят сложности лишь до некоторой степени, а потом сдаются и просто отключают его.

Если у вас есть опыт работы с SELinux (удачный или не очень), черкните мне пару строк на chris.linuxformat@gmail.com. Было бы особенно интересно услышать тех, кому SELinux помог отразить настоящую атаку на их систему. Ведь именно ради этого мы и гордим огород. **LXF**



► Это графическая утилита настройки SELinux в RedHat. Мы видим пользователей SELinux и роли, которые они могут принимать.

Где узнать больше

Книга Билла Маккарти [Bill McCarty] по SELinux (опубликованная в издательстве «О’Рэйли») – в некотором смысле наиболее полная работа, хотя она и вышла в 2004 г. и ее возраст неизбежно сказывается. Руководство пользователя по SELinux в RedHat (http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Security-Enhanced_Linux) – более практичное и современное. Также масса информации есть на сайте selinuxproject.org.



Наши эксперты помогут вам с любым приложением Linux!



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

Kickstarter, игры и Linux

Я действительно не хочу уничтожать Microsoft – это будет побочный эффект.
Линус Торвальдс

Начинаем пересчитывать по новой: чего не хватает Linux? *Photoshop* – пусть будет раз, хотя для профессиональных фотографов *GIMP* уже предоставил поддержку 16 и 32 бит на канал. *AutoCAD* – пусть будет два, хотя если вам не хватает бесплатного *DraftSight*, то есть платный *VariCAD*, *Bricscad* и даже *NX*. Игры... Стоп, а почему не хватает игр? Безусловно, есть проблемы с AAA-проектами, но...

В каком-то смысле истинным студентам-линуксоидам по сравнению с их более элементарными товарищами значительно проще учиться после выхода *Diablo III*, *Mass Effect 3* и *Legend of Grimrock* – при условии, естественно, что возиться с *Wine* они не хотят. Но число красочных приманок с отметкой «работает под GNU/Linux», зазывающих в виртуальный мир, растет катастрофически быстро. Знак «Поддерживаем Linux» уже вывесили создатели *Full Throttle*, *Grim Fandango*, *Wasteland*, *Fallout*, *Shadow Runs*, *Gabriel Knight* и *Space Quest*. Появились и новые, молодые команды, с проектами *The Banner Saga*, *FTL: Faster Than Light*, *Nekro* и *Legends of Eisenwald*. К сожалению/счастью, вся эта роскошь доступна не прямо сейчас, но этак через год придется собирать урожай. Бедные студенты-линуксоиды.

P.S. Вышла Ubuntu 12.04. Если в следующий релиз, как грозятся, вкрячат Wayland без альтернатив, это будет моя последняя Убунта. Благо есть пять лет на раздумье.
E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Изготавливать шрифт 66
Подлежит ли тиражированию ваш почерк? **Шашанк Шарма** предоставляет для этого *FontForge*.



Чирикать онлайн 70
IRC – горячее проектов свободного кода. **Майк Сондерс** научит вас непринужденно вступать в беседу, вписавшись в любую компанию.



Быть гуру Vim 74
При содействии **Джонатана Робертса** вы освоите самые мощные и продвинутые функции *Vim*. Программисты и писатели, это для вас.



Звучать музыкально 78
В руках у **Ника Вейча** Arduino приобщается к классике. Мелочиться не будем: даешь Бетховена!



Вникать в Erlang 82
Андрей Ушаков отбивается от множества задач, пользуясь средствами, предусмотренными языком.



Охранять сеть 86
Вам нужен удаленный и одновременно безопасный доступ к вашей сети? **Нейл Ботвик** настраивает VPN...



Печатать задания 88
Дмитрий Цымай и **Станислав Куценко** фабрикуют тестовые задания для студентов с помощью *LaTeX*.

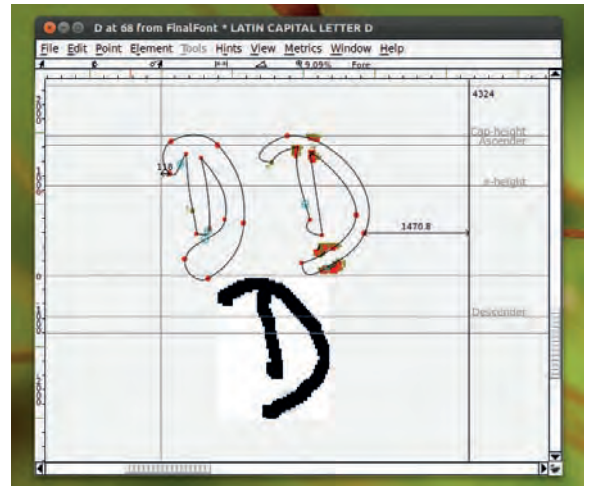
Пропустили номер?

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



FontForge: Ваш

Есть ли музыка внутри каждого из нас, остается под сомнением. А вот если внутри вас есть шрифт, Шашанк Шарма выведет его на свет.



» Автотрассировка облегчает жизнь — применяйте ее ко всем глифам, а за ручную работу беритесь только когда она не справляется.

символы шрифта, они же — глифы. Поскольку глифов мы еще не добавляли, все клетки в окне представления шрифтов перечеркнуты красным крестом.

Двойной щелчок на пустой клетке символа вызовет окно *Outline Glyph* [Эскиз глифа]. Здесь мы будем создавать и редактировать наши глифы. Заодно станут доступны другие два окна — *Tools* и *Layers*. Мы рассмотрим каждое из них в свою очередь, как только начнем использовать *FontForge*.

Шрифт делают глифы

В окне *Font View* щелкните *Element > Font Info* [Элемент > Данные о шрифте] и укажите имя шрифта. Там понадобится предоставить еще много информации, например, размер EM и т.п., но мы вернемся к этому позже. Закончив, нажмите OK.

Если у вас еще нет растровых изображений для символов, сначала создайте их — обратитесь к клеткам.

Для добавления глифа к шрифту дважды щелкните на пустой клетке глифа шрифта в *Font View*. Откроется окно *Bitmap Glyph View* [Просмотр растрового глифа], а также окна *Tools* и *Layers*.

Окно *Layers* состоит из трех слоев: *Foreground* [Передний план], *Background* [Фон] и *Guides* [Направляющие].

Большую часть работы мы будем делать в слое *Foreground*, и эта работа станет частью нашего шрифта. В фоновый слой мы импортируем изображение нашего глифа. Вы также можете переключиться на фоновый слой и проверить, правильно ли вы очертили глиф. Наконец, в слое *Guides* задаются направляющие линии, которые применяются ко всем глифам шрифта.



Наш эксперт

Шашанк Шарма
Пишет о свободном ПО уже более четырех лет, включая Linux.com. Он соавтор *Beginning Fedora*.

При работе с вашим любимым текстовым процессором вам даются на выбор всякие шрифты: Liberation, DejaVu и другие. А вдруг вам захочется создать собственный шрифт, да не просто шрифт, а отображение вашего же почерка? *FontForge* помогает создавать шрифты с нуля, и мы здесь покажем вам, как это делается.

Быть может, для мира пришло время решить, права ли мисс Марпл, считая, что ваш почерк вас выдает.

В репозиториях большинства дистрибутивов *FontForge* есть, и его установка — левое дело. К сожалению, не все дистрибутивы содержат последнюю версию. Хуже того, версии, предлагаемые некоторыми дистрибутивами, глючат и падают, не успеете вы и охнуть.

Если вы решите установить *FontForge* из исходных текстов, то процесс состоит из проверенной триады *./ Configure, make* и *make install*.

Интерфейс

Подобно *GIMP*, *FontForge* распределил всю свою функциональность по трем окнам. В отличие от *GIMP*, однако, его конструкция не подвергается столь обильной критике (если вообще критикуется). Эти три окна — *Tools* [Инструменты], *Layers* [Слои] и *Main Window* [Главное окно], также называемое *Font View* [Просмотр шрифта].

Запустив *FontForge*, нажмите кнопку Создать на диалоговом окне Открыть шрифта. Откроется главное окно, где показаны все

«FontForge распределил свои функции по трем окнам, как GIMP.»

ЛИЧНЫЙ ШРИФТ

Высота у глифов будет неодинаковая. Очевидно, высотой различаются заглавные и строчные буквы, но и некоторые строчные буквы возвышаются над другими, а есть и такие, что залегают ниже других. Большинство букв сидит на опорной линии. Хвостики строчных букв, простирающиеся ниже опорной линии (строчные р, у, ф и т. д.) называются Descender [нижний выносной элемент]. Наоборот, Ascender [верхний выносной элемент] определяет верхинки строчных букв (б, t, l и т. д.), превышающие X-уровень, который, в свою очередь, определяет высоту строчной буквы (н, а, м, и т. д.) над основной линией. Последний из этих уровней – высота Cap, т. е. высота букв верхнего регистра над опорной линией.

Выберите Guides в окне Layers и перейдите в окно Glyph View, которое вы открыли двойным щелчком на пустой клетке глифа. Теперь создадим все направляющие. В окне уже есть несколько линий – опорная и линии, ограничивающие ширину буквы слева и справа. Теперь зададим линии высот над опорной линией. Они будут доступны для всех символов шрифта.

Формирование глифов

Теперь войдите в меню File > Import [Файл > Импорт] в окне Bitmap Glyph и импортируйте растровые изображения для редактируемого глифа. Мы начнем с одного из простейших глифов, так что дважды щелкните на пустой клетке F. Импортируемое изображение масштабируется, и нам надо подогнать его размер. Щелкните Background в окне Layer и измените размер импортированного изображения, позаботившись, чтобы оно вписалось в наши свежееопределенные высоты.

Затем в окне Layer переключитесь на Foreground. Мы подошли к наиболее важному и трудоемкому аспекту создания шрифтов – обводке, или трассировке, контура глифа. Он же и самый

Определения

При работе со шрифтами и FontForge вы часто будете сталкиваться с такими терминами, как шрифт, символ, контур, сплайн, сплайн Безье, и т. д., поэтому разберемся, как понимать каждый из них.

» **Шрифт** Цифровой файл, включающий глифы для всех символов на клавиатуре.

» **Символ** Любой печатаемый знак с клавиатуры. Т. е. сюда включаются все буквы, цифры и значки. Одна и та же буква в верхнем и нижнем регистре – это два разных символа.

» **Глиф** Рисованное представление символа. При работе с FontForge нужно создать глифы для каждого символа. Причина, по которой одни и те же символы по-разному выглядят в разных шрифтах

(сравните DejaVu Serif и Liberation Serif) в том, что за ними стоят разные глифы.

» **Контур, или Путь** Каждый символ состоит из замкнутых контуров. Скажем, для символа А один контур очерчивает его внешнюю форму, а другой – треугольник внутри. Контур состоит из нескольких сплайнов.

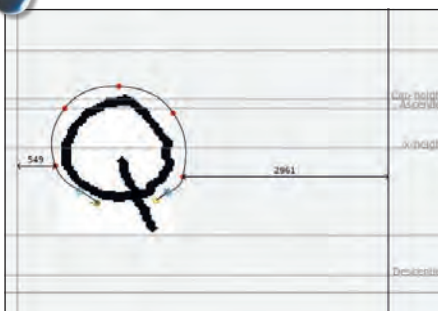
» **Сплайн** Определяется двумя конечными и несколькими управляющими точками. Управляющие точки определяют форму кривой, описываемой сплайном. Чем больше управляющих точек, тем более гибко задание формы. Сплайн без управляющих точек – прямая линия. При наличии одной управляющей точки эту прямую линию можно изогнуть, получив заглавное С.

ответственный: вы должны тактично разработать дизайн всех глифов, чтобы сделать их похожими на ваш рукописный почерк.

Хотя окно Tools предлагает различные инструменты для выбора, мы будем главным образом использовать только четыре: Pointer [Указатель] и точки типа Curve [Кривая], Corner [Угловая] и Tangent [Касательная]. При обходе контура обеспечьте, чтобы контур замкнулся, т. е. его начало совпало с концом – например, поставив конечную точку поверх начальной. Так, если вы начинаете

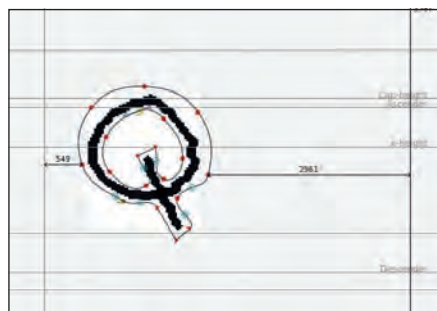


Шаг за шагом: Кривая, угловая и касательная точки



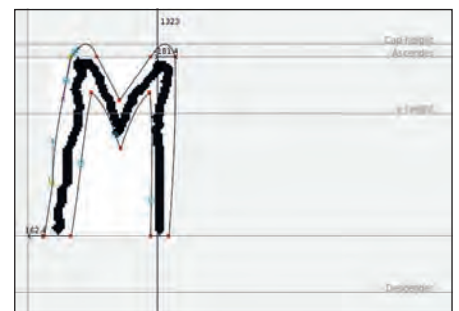
1 Curve — Кривая

Точку этого типа мы используем в местах, где линия изгибается. На рисунке мы используем точки типа Curve вдоль внешнего обвода с левой стороны и внутренней стороны обвода. Не используя точку Curve, можно создать кривую, перетаскивая отрезок, соединяющий две точки.



2 Tangent — Касательная

Для многих глифов вы обнаружите, что кривая переходит в прямую линию, или край; или прямая линия переходит в кривую, как хвостик нашей Q. Точки типа Tangent отмечают переход от кривой к прямой.



3 Corner — Угловая

Для некоторых глифов, таких как E и M, прямые линии круто меняют направление, часто на 90 градусов. Тут применяются точки типа Corner. Вместо них всегда можно использовать точку типа Curve, если вы хотите сделать углы глифа более округлыми.

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

свой путь с точки Tangent посреди вертикальной черты F, последняя точка должна прийти поверх точки Tangent, с которой вы начали. Если вы создадите незамкнутый контур пути, *FontForge* сообщит об ошибках в символе.

Просяним это, обведя контуром F. Начнем с точки Tangent посреди длинной ножки. Теперь выберите точку Corner в окне инструментов и щелкните на верхнем левом углу F, чтобы поставить точку там. Точка Corner у нас остается выбранной; щелкните на правом верхнем углу F, поместив там другую точку Corner, и поместите еще одну чуть ниже ее. Заглавная F – один из немногих глифов, которые можно обвести, ставя только точки Corner и Tangent. Продолжайте добавлять точки Corner везде, где меняется направление обвода символа. Закончив, поставьте точку над исходной точкой. Вот и все – вы только что успешно трассировали свой первый глиф!

При трассировке глифа, в зависимости от способа его рисования, можно полагаться на сочетание всех трех типов точек, или только один. Например, для буквы O – и строчной, и заглавной – достаточно точек типа Curve.

Не волнуйтесь, если контур неплотно прилегает к вашему глифу – просто добавляйте точки, пока не завершите обход. Закончив, переключитесь на слой Background. Контур, который вы только что трассировали, показан бледно-зеленым цветом. Убедитесь, что контур полностью замкнут.

Затем вернитесь на Foreground и переключитесь на инструмент Pointer. Теперь можно щелкать на добавленных точках и перетаскивать их, или щелкнуть на линии между двумя точками и перетянуть ее, если надо подправить контур. Наконец, перетащите линию отметки ширины, справа, поближе к обведенному контуру. Если вы находитесь на Foreground, *FontForge* покажет линии отметки ширины по обе стороны от глифа – они называются левым и правым ограничителем, L Bearing и R bearing, соответственно.

Скорая помощь

При регулировке ширины старой выдерживать одинаковую ширину для всех глифов. Можно менять только R Bearing – L Bearing фиксируется, как и опорная линия.

«Избегайте автотрассировки, хотя ручная и кажется муторной.»



► Измените в Element > Font Info > General размер Em с 1000 на 4096, если вы создаете шрифт TrueType.

Сохраните готовый глиф и откройте другой. Трассируйте все ваши глифы, используя ту же процедуру, и, наконец, настройте

правый ограничитель. Для каждого успешно трассированного глифа в Font View, вместо красного креста в его клетке появится этот глиф. Если трассировка каждого символа вручную вас раздражает,

альтернатива есть. Вас выручит инструмент Autotrace [Автотрассировка], который легко установить из репозитория ПО. После установки щелкните на Element > Autotrace [Элемент > Автотрассировка] в Bitmap Glyph View, и *FontForge* автоматически оттрассирует контур за вас. Мы, однако, рекомендуем: избегайте функции Autotrace, какой бы муторной ни казалась ручная трассировка.

Применив Autotrace, вы обнаружите, что на глифе поставлено слишком много точек – это помогает *FontForge* отследить в глифе все изъязны и все извивы и повороты. После Autotrace вам тут же захочется обратиться к инструменту Element > Simplify [Элемент > Упростить] – он объединит часть избыточных точек. Есть еще и Simplify More [Упростить далее] для дальнейшего контроля над точками, добавленными Autotrace.

Autotrace принадлежит к тем функциям, где ваш прогон может варьироваться, так как полностью зависит от способа рисования глифов. Главная проблема с Autotrace в том, что контуры получаются пересекающимися. Нам следовало отметить, что вы должны рисовать внешние контуры по часовой стрелке, а внутренние – против. Пересекающиеся контуры возникают, когда *FontForge* не может определить, идет ли обход по часовой стрелке или про-

Создание глифов для шрифта

Как ни здорово *FontForge* создает шрифты, он мало что может сделать без отдельных глифов, составляющих шрифт. Если вы хотите превратить ваш почерк в шрифт, есть два способа это сделать: либо написать все символы на бумаге, отсканировать их, а затем извлекать глифы из полученного изображения, либо использовать планшет для рисования глифов в цифровом виде.

Для нашего рукописного шрифта мы выбрали второй. Особых причин предпочесть один способ другому нет: мы просто порадовались, что наконец-то припомним к делу планшет, провалявшийся несколько лет.

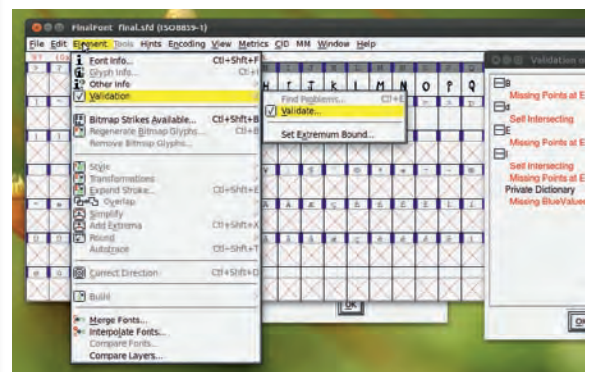
Сейчас большинство планшетов работают «из коробки» почти во всех дистрибутивах Linux, так что подключите шнур, запустите *GIMP* и рисуйте свои символы.

Лучше всего создать большой холст, способный вместить все ваши символы, а затем нарисовать их все сразу.

Если у вас не получатся красивые очертания, переключитесь на более тонкую кисть. А вот вам практический совет: пишите скорописью, так же, как на бумаге.

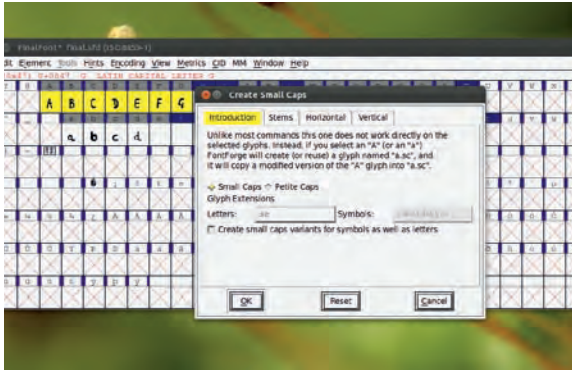
Аккуратно нарисовав все глифы, создайте отдельное растровое изображение для каждого. Выберите на панели инструментов Fuzzy Select [Умное выделение] и щелкните по глифу. Выберется весь глиф. Нажмите клавиши Ctrl+C и Ctrl+Shift+V, чтобы скопировать выбранный символ в новое изображение. Сохраните изображение как растровое (с расширением BMP).

В зависимости от способа их рисования, некоторые глифы, возможно, не полностью выберутся при щелчке на них – например, глиф для й. В этом случае воспользуйтесь клавишей Shift для выбора нескольких областей. Например, выберите основную часть й, а закорючку над ней – с помощью клавиши Shift. Тогда после копирования и вставки вы получите глиф целиком.



► Пользуйтесь Element > Validation > Validate Feature [Элемент > Проверка > Проверка функции] для проверки шрифта на наличие проблем.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.



» Для создания капители щелкните на **Element > Style > Add Small Caps**. Вместо изменения глифов будут созданы новые.

тив. А с Autotrace почти все наши глифы сообщили о пересечении контуров. В такой ситуации *FontForge* шрифт рисовать не умеет.

Для глифов с контурами и по часовой стрелке, и против (A, O, P, D и т.д.) пригодится функция **Correct Direction** [Верное направление], причем хорошей практикой будет применить ее ко всем глифам. В окне **Font View** выберите все оттрассированные глифы и нажмите **Element > Correct Direction**. При необходимости *FontForge* автоматически скорректирует направления во всех глифах.

Если вы предварительно создали направляющие линии и прилежно вгоняли все свои глифы в соответствии с ними, у вас получатся глифы унифицированного размера.

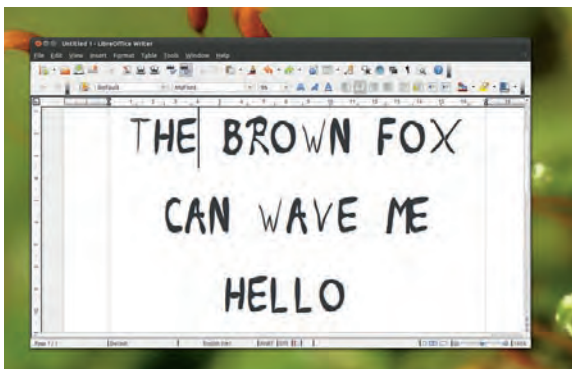
FontForge сохраняет шрифт в своем собственном формате **.sfd**. Рекомендуем сохранить шрифт после каждого изменения, внесенного в символ, хотя бы и тривиального.

Семейство шрифтов

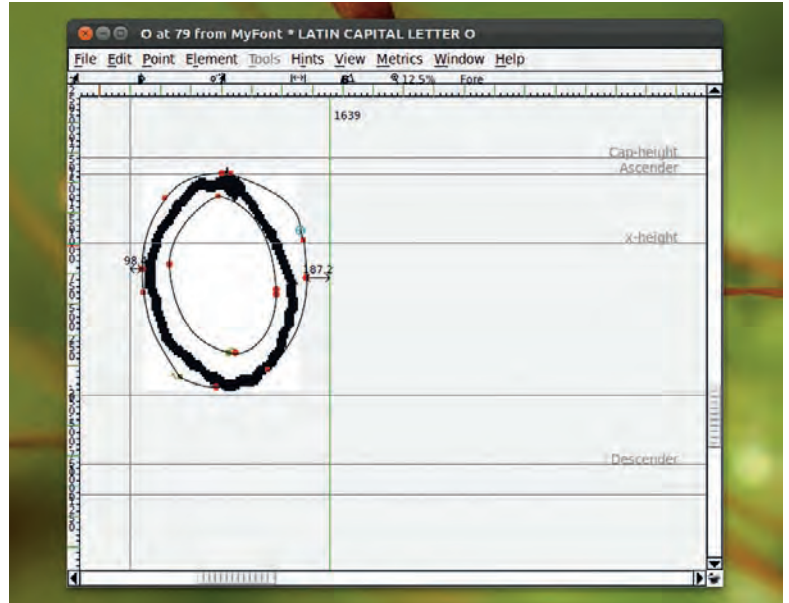
Но это еще не конец. Надо обучить наши глифы делаться жирными или курсивными. Чудесным образом это не принудит вас рисовать символы более толстой кистью или с наклоном. *FontForge* предлагает различные эффекты, пригодные для быстрого преобразования шрифта в разные стили, будь то полужирный или курсив.

Для начала выберите все глифы в шрифте и нажмите **Element > Styles > Change Weight** [Элемент > Стили > Изменить вес]. В этом диалоге несколько полей, и для выработки лучшего полужирного начертания пробуйте настройки, пока не добьетесь идеала.

Первое поле, **Embolden By** [Усилить на], определяет величину, на которую каждый штрих должен разрастись. Для выбора метода, которым будут утолщаться глифы, предусмотрены радиокнопки. Лучшая – **LCG**, так как она утолщает глифы, остав-



» Говорят, что картинка стоит тысячи слов, но верно ли это, если на ней показано всего лишь несколько?



ляя их в заданных границах. Закончив, сохраните шрифт – лучше под именем, отражающим примененный стиль. Скажем, если ваш базовый шрифт назывался **MyFont**, то жирный шрифт стоит назвать **MyFont-bold**.

Курсив, однако, создать гораздо сложнее. Есть люди, которые считают – и автор *FontForge*, похоже, с ними согласен – что если вам нужен идеальный курсив, его надо рисовать самим. Тем не менее, можно перейти на **Element > Styles > Italics** [Элемент > Стили > Курсив] и поиграть с настройками, обеспечив своему шрифту подходящий наклон. Закончив, сохраните шрифт под именем **MyFont-italics**, аналогично варианту с полужирным шрифтом.

В порядке альтернативы диалоговому окну **Italics** рекомендуем заглянуть в **Element > Styles > Oblique** [Элемент > Стили > Скос]. Единого преобразования в курсив для всех глифов сразу нет – приходится тщательно определять, как лучше закурсивить каждый глиф; а диалог **Oblique** требует только указать угол, на который вы хотите наклонить глифы.

Создание шрифта

Последний шаг заключается в создании устанавливаемого шрифта, и вы можете сделать это, нажав **File > Generate Fonts** [Файл > Создать шрифты]. Выберите тип шрифта, который хотите создать, будь то **TrueType**, **OpenType**, **SVG** и т.д., укажите имя и нажмите кнопку **Save** [Сохранить] в нижней левой части диалогового окна **Generate Fonts**.

Имя, указанное здесь, нужно только чтобы отличить файл шрифта. После установки шрифт будет отображаться с тем именем, которое вы ему дали в *FontForge* под **Element > Font Info**. Итак, если вы назвали шрифт **MyFont**, но сохранили его как **YourFont.ttf**, после установки вы найдете его в списке шрифтов **LibreOffice** как **MyFont**.

Для установки шрифта зайдите в каталог, где вы его сохранили, и дважды щелкните на нем мышью. Большинство дистрибутивов автоматически установит шрифт за вас. Если вам показалось, что шрифт нуждается в доработке, просто удалите его из каталога **~/fonts**.

А раз уж так легко создавать, устанавливать и удалять шрифты, вы можете генерировать шрифт на различных этапах развития и проверять, как он выглядит. **LXF**

» Не забудьте отрегулировать ширину для всех глифов, и постарайтесь сделать ее одинаковой, равномерности ради.

Скорая помощь

Следует рисовать внешние контуры по часовой стрелке, а внутренние – против.

Спасибо Джеймсу Полларду [James Pollard] за тему данного учебника.



IRC: Разговоры

Надоел Jabber? Бесят ограничения Google Chat? **Майк Сондерс** представляет Internet Relay Chat, самый лучший способ поболтать...



Наш эксперт

Майк Сондерс

Использует IRC добрых лет десять, а также имеет свой собственный канал. Где? Это секрет!

Представьте, что вы сидите за обеденным столом на 20 человек, и все они говорят. Обычно участники объединяются в подгруппы по четыре-пять человек и ведут отдельные беседы с небольшим количеством людей. Как достичь этого в Интернете? Напрашивается решение создать отдельные чаты, но оно не идеально. За обеденным столом, хотя каждый принимает участие в отдельном разговоре, можно услышать другие дискуссии и встретиться глазами. Разговоры можно объединить или перейти из одного к другому.

Очень немногие системы интернет-чатов умеют справляться с такой ситуацией, но самая лучшая из них – IRC, и она существует примерно с ледникового периода развития Интернет (первый сервер был создан в 1988 году). И хотя многие программы для чата приходят и уходят, IRC выдержала испытание временем по многим причинам:

- » Она идеально подходит для бесед сотен участников.
- » Выбор чат-клиентов огромен.
- » Можно автоматизировать ответы с помощью скриптов и ботов.
- » Создать свой собственный сервер IRC может каждый.
- » Изводящие дети и тролли обычно не знают об этом.

Трудно оценить, сколько людей используют IRC в каждый момент времени, но грубая оценка по одной сети – 500 000. По сути,

IRC работает следующим образом: кто-то запускает демон IRC (другими словами, сервер) на компьютере, к которому другие люди в Интернет могут подключиться через клиент IRC. Люди создают различные каналы (подобно комнатам в чатах) на сервере, и могут устанавливать различные уровни прав доступа для пользователей (например, оператор канала может запрещать нарушителям этикета вход на канал).

Можно найти каналы практически на любую тему, но IRC используется прежде всего для разработки открытого ПО, в качестве непосредственного отклика и как альтернатива почтовым рассылкам.

На нашем уроке мы покажем вам, как использовать IRC-клиент, создать имя пользователя, присоединиться к каналам и общаться с людьми. Мы также рассмотрим вопросы посложнее, такие как скрипты и боты, и в итоге вы будете чувствовать себя достаточно уверенно, чтобы рискнуть принять участие в общении, и впишетесь в любую сетевую беседу.

Существуют сотни IRC-клиентов, и в конце урока мы рассмотрим самые лучшие, но для начала воспользуемся *XChat* (www.xchat.org), наиболее популярной графической программой. Она устанавливается по умолчанию во многих дистрибутивах или должна быть доступна в репозитории пакетов.

Присоединимся к беседе

Запустите XChat, и первое, что вы увидите – окно (как показано на экранном снимке), где можно ввести имя пользователя и выбрать сеть для подключения. Сети являются коллекциями серверов IRC, и одна из самых популярных для открытого исходного кода и Linux – Freenode. Подключаясь ко Freenode, вы все время попадаете на разные сервера этой сети, но все каналы будут доступны (пользователи могут расширить пропускную способность сети IRC, настроив свой сервер и подключив его к сети, это немного похоже на одноранговый обмен файлами).

Итак, введем прозвище пользователя, которое еще не занято, и второй вариант, если оно занято. Затем выберите Freenode

в списке сетей и нажмите Подключиться. Появится главное окно, с информацией о подключении к серверу, а также диалоговое окно с запросом, к какому каналу вы хотите присоединиться. Это полезно для абсолютных новичков, но мы, вместо того, чтобы делать все через GUI, попрактикуемся с командами, которые работают во всех клиентах IRC. Итак, выбираем вариант «Ничего, я присоединюсь к каналу позже» и возвращаемся в главное окно *XChat*.

Вы заметите, что внизу – рядом с вашим именем пользователя – есть текстовые поля для ввода. Тут вы набираете сообщение для разговоров, а также команды. В мире IRC, команды начинаются с косой черты и отправляются не в сам чат, а непосредственно на сервер IRC. Например, если вы видите сообщение о том, что имя пользователя, которое вы выбрали, уже используется или зарегистрировано, вы можете попробовать другое с помощью следующей команды:

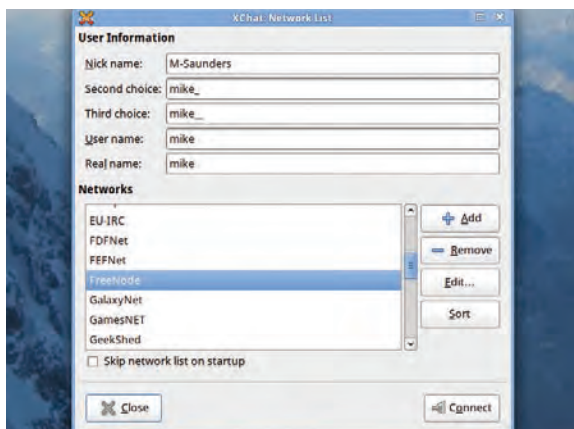
```
/nick mynewnickname
```

С помощью этой команды вы отправляете сообщение на IRC-сервер и говорите, что вы хотите изменить свое имя пользователя, а сервер будет пытаться выполнить операцию, выдавая результаты в главном окне вывода. На данном этапе мы не можем ни с кем разговаривать, так как мы только установили соединение с сервером; чтобы найти готовых к беседе пользователей, нужно войти на канал с помощью команды `/join`, как показано далее:

```
/join #ubuntu
```

Отметим, что в большинстве случаев названия каналов начинаются с символа «решетка», и с этой командой мы пытаемся присоединиться к каналу, где пользователи Ubuntu обсуждают

» Запустив *XChat*, вы увидите экран для выбора имени пользователя и сети.



В строю

Исследуем Xchat

Список разговоров

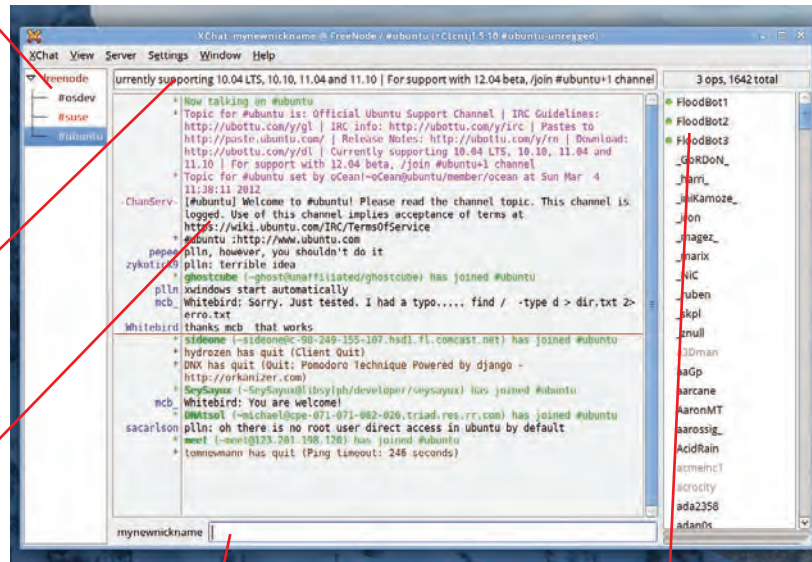
Здесь вы можете переключаться между серверами, к которым вы подключены, каналами, на которые вы заходили, и личными разговорами. Когда название канала отмечено красным, это означает, что там возникла новая активность (например, кто-то зашел).

Название канала

Здесь вы увидите название канала, а нередко и важную информацию для новых пользователей – например, сопутствующие каналы и правила поведения.

Главное окно

Тут все и происходит. Имена пользователей отмечены синим, а сообщения – черным. Сообщения сервера фиолетовые, зеленый цвет используется для информации о входах новых пользователей, а коричневый – для выходов. Красная линия отмечает начало сообщений с момента вашей последней активности.



Поле ввода текста

В нем вы вводите свои сообщения и команды. Здесь включена функция предистории – пользуйтесь стрелками вверх и вниз для прокрутки предыдущих сообщений.

Список пользователей

Все пользователи на канале. Зеленым отмечены операторы канала, серым – неактивные пользователи. Щелкните правой клавишей на имени пользователя для дополнительной информации.

Скорая помощь

Возможно, гуляя по IRC, вы уже слышали термин 'netsplit'. Это происходит, когда между двумя и более серверами в одной сети прерываются коммуникации. Если пять пользователей находятся на сервере А, а другие пять – на Б, и связь между ними прервалась, сеть разбивается на две подсети. Пять пользователей на сервере А могут вести разговор только между собой, пользователи на сервере Б находятся в такой же ситуации.

дистрибутив. Это один из самых оживленных каналов в Интернет – обычно на нем присутствуют от 1000 до 2000 пользователей, хотя многие из них будут просто «висеть» (т.е. заниматься другими делами). Можете попробовать другие каналы, такие как #orepense или #osdev (для разработки операционной системы).

По картинке сверху, с объяснениями, вы получите представление о главном экране XChat. Попробуйте начать разговор – за-

дать вопрос, например: «Какие самые лучшие ноутбуки для работы в Ubuntu?» В зависимости от активности на канале, вы можете получить различные ответы, и некоторые из них могут появиться в красном цвете. Это означает, что было упомянуто ваше имя пользователя. На оживленном канале IRC может вестись 10 или больше разговоров одновременно, и для людей, которым необходимо следовать разговору, сложно, если имена пользователей

Поиск нужной сети

Существуют тысячи IRC-сетей, большинство из которых невелики, всего с одним оператором, и созданы под конкретные потребности, но в основном активность кипит в большой четверке:

» **Freemote** (<http://freemote.net>) – проектный центр, основанный на одноранговых коммуникациях. Иными словами, место для совместной работы над проектами, подобными проектам свободного программного обеспечения. Здесь более 40000 каналов, в том числе #Ubuntu, #Debian, #python, #Inkscape, #Scribus и #Linux.

» **Efnet** (www.efnet.org) – крупная сеть с более чем 35 000 пользователями, охватывающая широкий спектр вопросов, включая гаджеты и игры. Известные каналы включают #iphone, #StarCraft и #PS3. С годами Efnet приобрела репутацию «андерграунда», то есть такого места, где можно отыскать пиратское программное обеспечение и встретить группы хакеров.

» **IRCnet** (www.ircnet.org) – одна из наиболее хорошо регламентированных сетей IRC, где правила строго соблюдаются и пользователям, как правило,

не разрешается запускать ботов. Здесь нет особой специализации, но самые оживленные ее каналы работают на определенных языках (польский, финский, японский и т.д.).

» **QuakeNet** (www.quakenet.org) – появилась в 1997 году как сеть для фанатов Quake, и до сих пор сохраняет свою специализацию на видеоиграх, с тысячами каналов, охватывающими широкий спектр консолей и игр. Это хорошее место, чтобы найти игроков онлайн-игр; например, на канале #S.pcw – для Counter-Strike: Source.

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

не используются. Так что, если Боб и Билл разговаривают о ноутбуках, а вы хотите присоединиться, но все остальные говорят про Gnome, то введите нечто вроде:

Боб: Я слышал, что ноутбуки ThinkPad вроде приличные

Обратите внимание, что *XChat* (и многие другие клиенты IRC) поддерживает автодополнение при вводе имени пользователя, поэтому вы можете просто ввести несколько первых символов и нажать Tab для ввода остальной части имени.

К дополнительным каналам можно присоединиться командой `/join`, а решив уйти с канала, введите `/part`. С командой `/quit` будьте осторожны: она разрывает все соединения с сервером(ами) IRC, и вы уйдете со всех каналов сразу. Но можно и опять вернуться:

`/server irc.freenode.net`

Чтобы получить информацию о пользователе, введите

`/whois имяПользователя`

Также можно узнать, какой IRC-клиент они используют, с помощью команды `/стр имя пользователя version`; но некоторые клиенты скрывают эту информацию. Если кто-то вас раздражает, и вы не хотите получать больше сообщений от них, введите `/ignore имя пользователя all`. Еще одна полезная команда – `/me`, которую вы могли видеть в других местах интернет-общения для обозначения действия. Если вы введете

`/me ест биг-мак`

и ваше имя пользователя ОдинТип, то все остальные увидят это в своих IRC-клиентах как

*** ОдинТип ест биг-мак**

Подробный список всех каналов, доступных в сети, можно посмотреть по команде `/list` (вывод будет в окне сервера, доступном сверху слева), но учтите, что некоторые из них требуют зарегистрированных имен пользователя для входа. Некоторые каналы очень замкнуты и на новичков смотрят косо, а некоторые разработаны специально для облегчения людям жизни по конкретной теме. Но вы не робейте: руководствуйтесь основными правилами поведения в IRC (см. врезку), и все будет хорошо.

В IRC возможны и частные беседы один-на-один – для этого используйте команду `/msg`, например, так:

`/msg имяПользователя Давай поговорим об этом между нами`

Если пользователь отвечает, его имя появится в списке (вверху слева). Щелкнув по нему, вы перейдете на этот частный разговор. Вы даже можете отправлять другим пользователям файлы, командой `/dcc [напрямую клиент-клиент]`, например:

`/dcc send имяПользователя имяФайла`

Здесь, замените **имяПользователя** на имя получателя файла. А пользователь на другом конце заберет его с помощью

`/dcc get получить имяПользователя имяФайла`

Ему здесь нужно ввести имя отправителя.

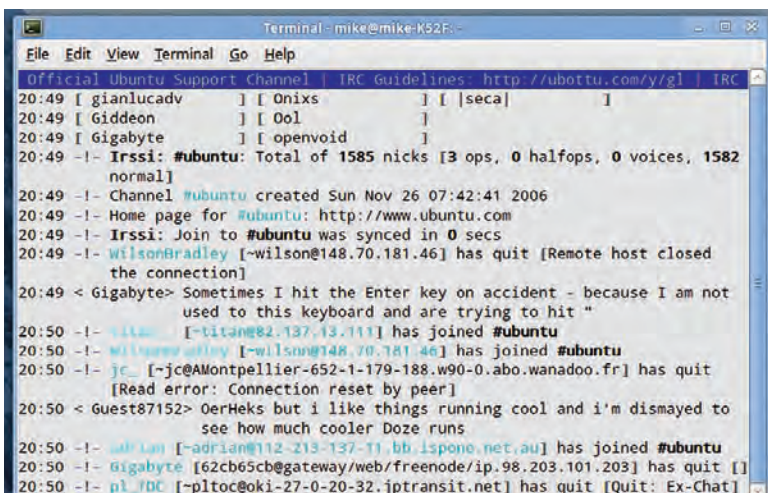
Скрипты и боты

Возможности *XChat* можно расширить скриптами, которые добавляют клиенту функциональность или изменяют его поведение. Официальный список можно найти на сайте http://xchat.org/cgi-bin/script_list.pl – но на самом деле, он довольно неудобен, поскольку нет возможности просматривать по популярности и рейтингу. Ресурс несколько получше, хотя и с меньшим выбором скриптов, можно найти на <http://digdilem.org/irc>. Установим, например, скрипт `colored_nicks_everywhere` [цветные ники везде], чтобы подчеркивать имена пользователей в теле сообщения – это облегчает ведение разговоров на оживленных каналах.

Во-первых, скачайте `colored_nicks_everywhere.py` (со второго из названных сайтов) в свой домашний каталог. Затем в *XChat*, перейдя в Окно > Плагины и скрипты, нажмите на вариант Python (потому что имя скрипта заканчивается на `.py`), а затем загрузите его. После чего выберите загруженный скрипт. Вернувшись в главное окно *XChat*, мы увидим обратную связь в виде `Loaded v0.6.colored_nicks_everywhere v0.6`, и по мере появления имен пользователей скрипт берет их и выделяет в главном окне обсуждения.

Вы найдете массу скриптов, которые подключаются к вашим музыкальным приложениям и делают объявления в канале чата, когда вы играете новую песню. Это хорошо работает в кругу друзей, интересующихся новинками, но, конечно, не стоит такое делать на оживленных технических каналах. Один из примеров подобных скриптов – спамер *VLC NP* от SteelGolem (NP означает «сейчас играет»), передающий в IRC-канал музыку, которую вы слушаете в *VLC*. После установки вы увидите сообщение:

Вы найдете массу скриптов, которые подключаются к вашим музыкальным приложениям и делают объявления в канале чата, когда вы играете новую песню. Это хорошо работает в кругу друзей, интересующихся новинками, но, конечно, не стоит такое делать на оживленных технических каналах. Один из примеров подобных скриптов – спамер *VLC NP* от SteelGolem (NP означает «сейчас играет»), передающий в IRC-канал музыку, которую вы слушаете в *VLC*. После установки вы увидите сообщение:



> Irssi запускается из командной строки и очень гибок благодаря системе скриптов.

Создание своего имени пользователя

Задать вопрос в канале поддержки вы можете под любым псевдонимом. Но если вы хотите стать узнаваемым на определенных каналах, вам требуется имя пользователя, которое вы можете использовать всегда. Это стало возможным в большинстве крупных сетей с помощью сервисов – то есть ботов, к которым можно обратиться и кото-

рые поддерживают сети в порядке. Один из таких ботов – сервер имен пользователей или “nickserv”. Вы можете зарегистрировать ник на Freenode следующим образом:

`/msg nickserv register пароль email@address.com`

Введите здесь свой пароль и адрес электронной почты, и вы получите письмо с объяснением про-

цедуры подтверждения регистрации. В следующий раз вы сможете авторизоваться на IRC с помощью:

`/msg nickserv identify пароль`

Вместо «пароль» подставьте свой. Это полезно, потому что, если кто-то временно употребил ваше имя пользователя, пока вас не было в сети, им придется взять другое.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Наберите `/np` и спамьте в чат о том, что играет VLC.

Когда VLC загружен и проигрывает музыку, при вводе `/np` все участники в канале получают следующее сообщение:

* ОдинТип пр: Zelenka Concerto in G - 3 Mvt.mp3

IRC-боты – это автономные программы, которые подключаются к IRC-сетям и каналам и выполняют автоматические действия. Они могут периодически выводить информацию на основе таймера, мониторить активность пользователей или просто отвечать на входящие сообщения. Существует много различных ботов – и сложных, и действительно простых; и один из лучших для того, чтобы научиться работе с ними – *Phenny* (<http://inamidst.com/phenny/>). Скачайте файл `.tar.bz2`, распакуйте его и перейдите в получившийся каталог.

Затем запустите `.phenny` – создастся файл `.phenny/default.py` в вашем домашнем каталоге – и отредактируйте его, чтобы изменить настройки по умолчанию. В частности, вам уж непременно захочется изменить первые четыре строки, где указаны имя пользователя для бота, сервер, к которому он должен присоединиться, каналы, к которым он должен присоединиться, и его владелец (то есть ваш псевдоним в IRC). Теперь запустите `.phenny` снова, и вы увидите вывод различной информации по соединению при подключении.

Пользователи на том же канале, где живет бот *Phenny*, могут взаимодействовать с ним с помощью команд. Например:

- » `.c 14*73` отображает результат вычисления;
- » `.g kittens` ищет котят в Google;
- » `.wik Linux` показывает первую строку из Википедии для Linux;
- » `.in 5` показывает напоминание для вашего имени пользователя через 5 минут.

Есть также и другие команды; вы найдете их в документации. Если вы хотите создать более специализированного бота для кон-

Журналы чата

Хотите сохранить свои беседы, чтобы почитать на досуге? В *XChat* перейдите в меню Настройки > Настройки и регистрации (в Чате) слева. Установите флажок «Включить протоколирование разговоров на диске», и в будущем вы найдете текстовые файлы журнала в `.xchat2/xchatlogs/` в вашей домашней директории. С помощью *PISG* (<http://pisg.sourceforge.net>) можно создать красивые HTML-страницы и графики, показывающие активность на IRC-канале, на основе файлов журнала.

кретного канала, придется замарать руки кодом. У нас есть руководство по написанию собственного бота IRC на Perl: www.tuxradar.com/content/code-project-build-irc-bot. Оно позволяет изготовить бота с пользовательскими командами и ответами. Другой вариант заключается в использовании «платформы» типа *Eggdrop* (www.eggheads.org), которая для реализации своих возможностей использует Tcl-скрипты и модули C.

Под занавес, пройдемся по некоторым другим популярным IRC-клиентам.

Для поклонников KDE отличным выбором является *Konversation* (<http://konversation.kde.org>), так как это очень зрелый клиент, под завязку набитый функциями. Те, кто склонен жить только в командной строке, могут использовать *Irssi* (www.irssi.org), который имеет широкий спектр дополнительных скриптов. Еще один популярный текстовый клиент – *BitchX*, хотя он несколько лет не обновлялся и не лишен проблем с безопасностью. Чтобы узнать больше, перейдите на www.freecode.com и выполните поиск по “IRC client”. Наслаждайтесь! **LXF**

Четыре шага к популярности на IRC

Большинство правил в IRC такие же, как и везде: не оскорблять людей, не начинать войнушек, не спамить и не размещать ссылки на фотографии Lolcat, если только это не абсолютно необходимо. Тем не менее, у IRC есть характерные ляпы, которых следует избегать.

» **Скройтесь на некоторое время** – понаблюдайте несколько минут, как ведут себя собеседники, прежде чем писать что ни попадя, если только это не срочный вопрос в поддержку канала. Это件 полезно для определения общего настроения, чтобы не вызвать сцены, сказав что-то неподходящее.

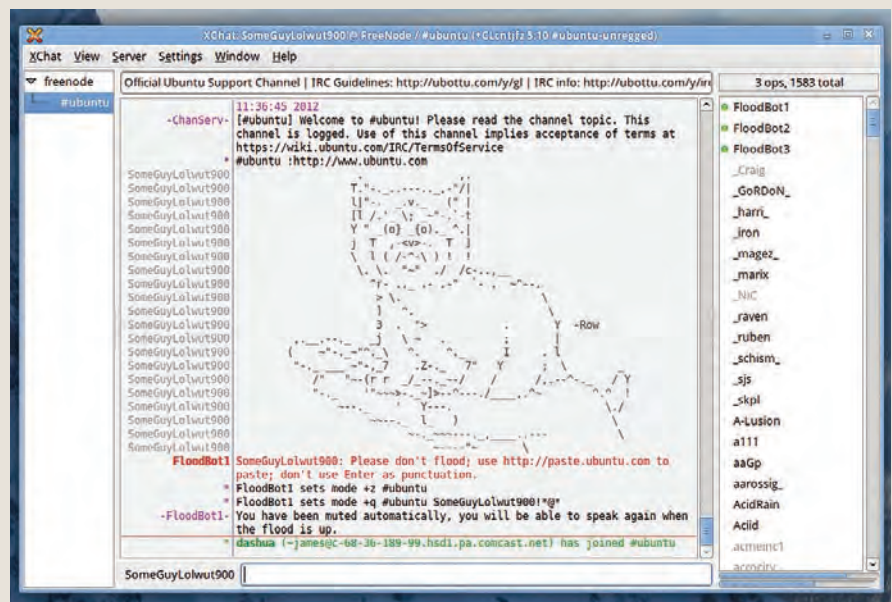
» **Флуд** – если вам нужно процитировать кому-то большой кусок текста, например, код или файл конфигурации, не копируйте его на канал. Это завалит экраны всех пользователей (когда они, скорее всего, говорят о другом), а в некоторых сетях вас даже удалят за слишком длинное сообщение. Вместо этого следует пользоваться сайтами вроде www.pastebin.com, где можно вставить кусок текста и получить на него URL-ссылку.

» **Не орите** – **Caps Lock может считаться ПОКАЗАТЕЛЕМ КРУТОСТИ** в некоторых интернет-кругах, и иногда приемлемо для привлечения внимания, но если вы напишете таким образом все слова, люди подумают, что вы кричите.

» **Не используйте цвета** – Возможно, вы нашли настройки цветов в вашем IRC-клиенте и хотите оживить скучный канал, но большинству это не нравится и в некоторых IRC-клиентах может выглядеть абракадаброй. Большинство каналов толерантны

и не будут вас сурово наказывать за нечаянные ошибки, но как знать – у операторов бывают плохие дни, когда им хочется на ком-то сорвать злость. Если вас шуганули, это не значит навсегда: вы просто временно отстранены. Не стоит возвращаться

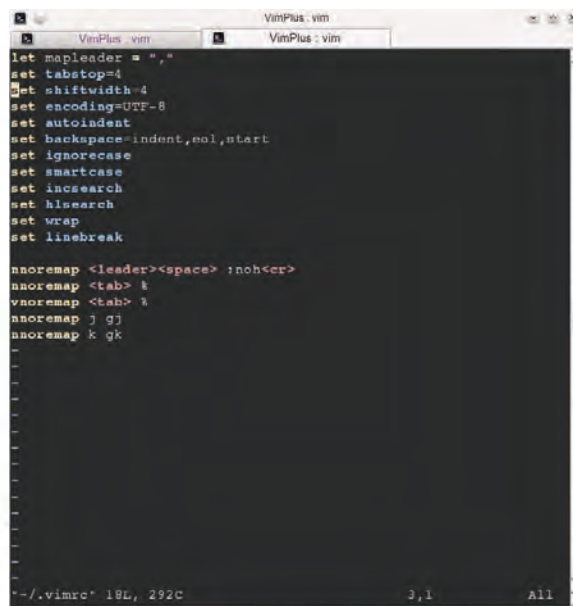
сразу: попейте чайку и приходите минут через пять. Если вас изгнали с канала, тут мало что можно сделать, кроме как умолять оператора канала – но ведь это ниже вашего достоинства, верно? Им же хуже, в конце концов...



» Вот что будет, если мы učinим флуд в #ubuntu ASCII-котятми: отсылка к *Pastebin* и автоматическое лишение голоса после пары первых строчек (то есть остальное никто не увидит).

Vim: ФУНКЦИИ

Учитесь писать код быстрее и лучше – Джонатан Робертс покажет вам некоторые расширенные функции Vim для высшего пилотажа.



» Этот пример файла .vimrc демонстрирует ряд «насыщенных» добавок для писателей и программистов.



Наш эксперт

Супер-скрытный Джонатан Робертс копается в Linux с детских лет. Иными словами, не так уж долго.

В LXF156 мы познакомили вас с Vim, продемонстрировав достаточно его функций, чтобы с ним управляться. Знание основ Vim полезно сама по себе, поскольку вы заодно будете знать, как использовать Vi, текстовый редактор, установленный по умолчанию практически во всех Unix-подобных системах.

Но Vim способен на гораздо большее, и голые основы сами по себе не объясняют, почему многие люди, программисты и писатели, обожают данный текстовый редактор. На уроке этого месяца мы решили просветить вас, рассмотрев ряд продвинутых функций Vim. Давайте начнем с идеи, что команды Vim составляют язык редактирования текста.

В LXF156 мы показали, как перемещаться по тексту документа. Если вы помните, клавиши **h**, **j**, **k** и **l** перемещают влево, вниз, вверх и вправо, **0** переместит вас в начало строки, а **\$** – в конец строки.

В Vim эти действия называют «подвижки [motion]». Команд-подвижек еще много. **w**, например, перемещает к началу следующего слова, **b** – к началу предыдущего слова, а **(and)** двигает вперед и назад одно предложение. Чтобы увидеть все команды-подвижки, скомандуйте в Vim: **help motion**.

Есть, однако, другая категория умных команд, которых мы в прошлый раз не рассматривали. Они известны как операторы [operator], и, как нетрудно догадаться, они изменяют части текста документа. Например, **c** изменяет, **d** удаляет, а **u** «выдергивает [yank]», или копирует, выделенный текст. Умны эти команды

благодаря тому, что они используются не сами по себе, но в сочетании с подвижкой.

Для удаления целого слова поместите курсор на начало этого слова, а затем введите **dw**. Если вы решите, что вам не нравится последнее написанное предложение, вы можете изменить его, введя **ц**.

Более того, вы не ограничены удалением одной строки, слова или любой другой используемой подвижкой. Все команды могут предваряться «счетчиком», который определяет, сколько раз выполнять это действие. **d2w** удаляет два слова, а **ц3** изменит следующие три предложения.

Вы даже можете комбинировать счетчик и подвижку без участия оператора: так, **4w** переместит курсор на четыре слова вперед. Поскольку команды можно объединять, Стив Лош [Steve Losh] описывает интерфейс команд Vim как подобие языка. Операторы – как глаголы: они «делают» слова и описывают то, что мы хотим, чтобы случилось; подвижки – как существительные: они задают объект, на который мы хотим повлиять глаголом. Наиболее важная идея, которую можно отсюда почерпнуть – то, что любую подвижку можно сочетать с любым оператором. Это означает, что, потихоньку изучая различные подвижки и операторы, на деле вы изучаете мощный язык редактирования текста.

Нужно только помнить общую форму команды Vim: **<счетчик><оператор><текстовый объект или подвижка>**. Это классная идея, которая вводит в самую суть причины, почему Vim так любим теми, кто потратил время на его изучение.

«Стив Лош описывает интерфейс команд Vim как подобие языка.»

ДЛЯ КРУТЫХ

Vim для программистов

Для целого поколения программистов, Vim и аналогичные простые (но мощные) текстовые редакторы были вытеснены интегрированными средами разработки (IDE). Эти среды поставляются с массой функций, предназначенных для повышения продуктивности кодирования, включая подсветку синтаксиса, файловые браузеры, менеджеры проектов и интегрированные компиляторы – все, что нужно для написания кода.

Они также раздуты и зачастую не имеют продвинутых функций редактирования текста, придающих Vim такую мощь. Однако вы сможете насладиться преимуществами обоих, настроив Vim.

Встроенные приятности

Во-первых, встроенные приятности. В нашей установке Fedora Vim идет с поддержкой подсветки синтаксиса и с 534 определениями различных

языков. Если вы используете экзотический язык, не охваченный этим набором, или недовольны предоставленным файлом, можете заглянуть на www.vim.org/scripts – там много новых и альтернативных файлов синтаксиса.

Также имеется автодополнение. По умолчанию, ввод нескольких букв слова и последующее нажатие **Ctrl+n** вызовет список возможных совпадений из остальной части документа. Прокрутите эти совпадения с **Ctrl+n**, а затем продолжите ввод для завершения слова. Это отличный способ быстро завершать функции и имена переменных, без опечаток.

Модули расширения

Затем, есть модули расширения, которые вам захочется установить. **surround.vim** добавляет новый текстовый объект, **s**, и вы можете применять его для изменения окружающих кавычек, тэгов, квадратных

или круглых скобок. Чтобы его использовать, наберите **cs'''** – окружающие двойные кавычки изменятся на одиночные; **ysiw** добавят тэги усиления текущего слова, а **ds** удалит их.

Еще один отличный модуль – **taglist.vim**. Он анализирует файлы кода, определяет функции, классы, перечисления и т. д., а затем составляет индекс, что позволяет быстро перейти к их определению в файле. Если у вас есть несколько буферов открыты, все они будут доступны из отдельного списка.

Два других модуля настоятельно рекомендуются для кодирования. **NERDTree** предоставляет встроенный браузер файлов, облегчающий управление файлами вашего проекта и открытие новых буферов. **snipMate** предоставляет готовые фрагменты для многих языков, в том числе циклы, функции и определения классов, чтобы сэкономить вам печатание.

Другие объекты

Тем не менее, при всей крутости рассмотренных команд они способны малость досадить: желая удалить слово, нужно тщательно позиционировать курсор на первой букве этого слова. Правда, при столь гибкой системе подвигек Vim (о которой мы скоро узнаем больше), это не самая большая проблема в мире.

Но Vim – это Vim, и его авторы создали способ смягчить и эту проблему. Vim имеет ряд «текстовых объектов», которые работают как подвигки, но действуют на весь объект, а не в направлении движения.

Все эти текстовые объекты начинаются либо с **a**, либо с **i**. Они указывают, желаете ли вы работать «вокруг» объекта, то есть прихватывать окружающие пробелы, или «внутри» объекта, то есть пробелы игнорировать.

Осталось узнать, с какими объектами можно работать подобным образом. Есть **w** для работы над словом, **s** – над предложениями, **p** – над абзацами, а также все виды скобок и кавычек.

Рассмотрим такой кусок кода JavaScript:

```
function vimRules(alpha, beta) { }
```

При желании изменить параметры, передаваемые в функцию, вы можете поместить курсор в любом месте в скобках и набрать **ci**, для «изменения в скобках». Текущие аргументы будут удалены, скобки останутся на месте, а Vim перейдет в режим вставки, готовый, чтобы вы начали печатать.

Или, если вы работали над сочинением или статьей и подумали, что предыдущее предложение было несколько многословно, вам нужно всего лишь поместить курсор в любом месте предыдущего предложения и набрать **cas**, и предложения не станет.

Перемещение

Теперь, зная, как язык Vim работает, и изучив различные подвигки, текстовые объекты и операторы, вы быстро обнаружите, что можете редактировать документы такими способами, которыми вам раньше и в голову бы не пришли. Но даже с текстовыми объектами вы все равно должны быть в состоянии установить курсор на нужный объект быстро, если намерены редактировать эффективно. Подвигки вроде **te**, с которыми мы уже сталкивались, позволяют легко перемещаться в широких направлениях. Для более точных перемещений, однако, удобнее будет использовать операторы поиска Vim.

Ввод **/** переведет Vim в режим поиска: Vim будет ожидать ввода строки, которую вы ищете.

Нажатие **Return** переместит курсор на первое совпадение, а повторное нажатие **n** переместит курсор на последующие.

Режим поиска Vim – весьма эффективный способ быстрого перемещения по документу, но, как и большинство опций Vim по умолчанию, может быть изменен и улучшен посредством манипуляции некоторыми из многих доступных настроек. »

«При всей крутости команд, они способны малость досадить.»

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

Vim для писателей

Настройки *Vim* по умолчанию не слишком подходят для прозаиков. Это не значит, что стандартные клавиши редактирования текста бесполезны — вовсе нет; но есть несколько неприятностей, которые лучше бы исправить, прежде чем серьезно заняться писательством в *Vim*.

В основе многих из этих проблем лежит манера *Vim* трактовать абзацы как длинные строки. В частности, это мешает использовать клавиши курсора по умолчанию для перемещения по линиям «экрана», и в результате текст заламывается посреди

слова. Обе эти проблемы решаются добавлением нескольких записей в файл `.vimrc`. Чтобы справиться с первой, переназначьте команды `gj` и `gk` как `j` и `k`:

```
nnoremap j gj
nnoremap k gk
```

Теперь при наборе `j` или `k` в Обычном режиме курсор будет двигаться вверх или вниз на одну строку экрана. Вы также можете переназначить `$` и `O` для перемещения курсора в конец и начало строки экрана. Для решения проблемы заламывания нужны такие параметры в файле `.vimrc`:

```
set wrap
set linebreak
```

`wrap` велит *Vim* заламывать длинные строки текста на краю видимого экрана, без добавления символа перевода строки.

`linebreak` велит *Vim* заламывать текст только на символах, перечисленных в опции `breakat`. По умолчанию сюда включаются пробелы, знаки табуляции и знаки препинания.

После этого писателям уже не придется страдать от разорванных слов в конце строки.

В *Vim*, попробуйте выполнить `:set incsearch` и поискать некий текст. Вы заметите, что вхождения будут выделяться по мере ввода. Вы можете использовать это, чтобы добраться до места в тексте еще быстрее, набрав лишь кусок слова, достаточный для его однозначной идентификации. Увидев, что нужное слово выделено, нажмите **Return** и выполняйте желаемые команды редактирования.

Поиск можно уточнить, выполнив команды `:set ignorecase` и `:set smartcase`. Первая делает поиск без учета регистра, а вторая отменяет результаты первой, если в строке поиска присутствуют заглавные буквы. Таким образом, поиск “bob” обнаружит “Bob”, “BOB” и “BoB”; а поиск “Bob” — только “Bob”.

Vimrc

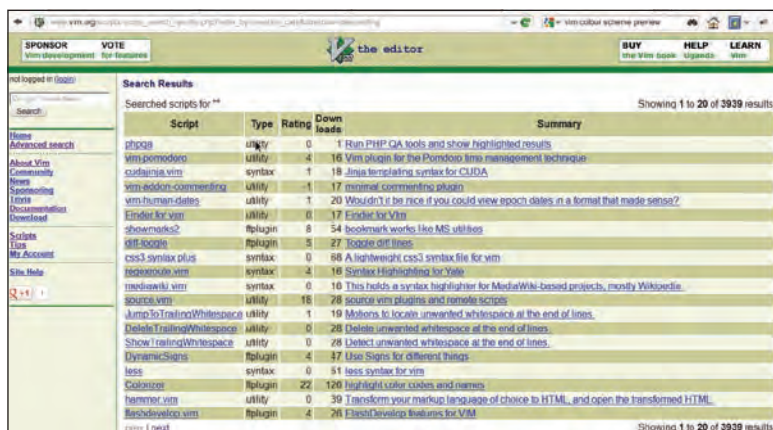
Vim имеет много, ну очень много настроек. Вы можете просмотреть их все, выполнив `:options`. По мере просмотра, нажатие на **Return** на строке ‘set’ будет переключать опции, а нажатие **Return** на строке описания даст вам полное описание данного параметра.

Продвигаясь дальше и узнавая больше о различных параметрах, которые вы хотите включать при каждом запуске сеанса *Vim*, можно настроить *Vim*, чтобы любимые настройки сразу вошли в действие.

Для этого *Vim* предоставляет файл `/home/<user>/.vimrc`. Это ваш личный файл настроек. В каждой строке помещаются отдельные команды, которые вы могли бы печатать после двоеточия в Обычном [Normal] режиме *Vim*. Например, если вы хотите включить указанные выше параметры поиска по умолчанию, ваш файл `.vimrc` будет выглядеть так:

```
set incsearch
set ignorecase
```

➤ Сайт *Vim* при-
ютил сотни скрип-
тов, позволяющих
настраивать *Vim*
самым чудесным
образом.



set smartcase

Заметим, что для ввода настроек в действие следует перезапустить *Vim*.

Забавный способ получить представление о том, какие параметры полезно включить в файл `.vimrc` — погладеть на других людей. Многие вывешивают свои настройки в Интернет, так что быстрый поиск по `vimrc` добудет много результатов. Вы также можете ознакомиться с двумя врезками, *Vim для программистов* и *Vim для писателей*, чтобы увидеть, какие параметры стоит включить, если вы попадаете в любой из этих категорий.

Дерготня

Мы уже рассмотрели самые распространенные функции редактирования и передовые приемы *Vim*, которые делают его эффективнее и мощнее. Единственное, о чем мы пока не говорили, однако, это копирование и вставка.

Копирование и вставку мы считаем сами собой разумеющимися в графических приложениях. Вы выделяете текст мышью и выбираете соответствующую опцию в меню Правка. Но *Vim* — консольное приложение, и в нем эти действия выполняются немного по-другому.

Во-первых, основы. Копирование в *Vim* называется «выдергиванием [yanking]» и выполняется клавишей **y** в Обычном или Визуальном [Visual] режиме. Вставка осуществляется по клавише **p**. Вырезание в *Vim* — это все равно что удаление, и вырезать текст можно любой из команд удаления.

Вместо того, чтобы сохранять выдергивание, вставку и вырезание в одном месте, *Vim* имеет несколько регистров. Имена у них от `a` до `z`, и вы в любой момент времени можете получить 26 различных кусочков текста.

Регистры работают, предвзяв одну из указанных операций строкой “a, где” ‘a’ — имя регистра. Итак, чтобы скопировать слово в регистр ‘b’, вы должны ввести “**byw**” в Обычном режиме. Для вставки этого текста введите “**bp**”.

Имя регистра указывать не обязательно. Если этого не сделать, *Vim* просто сохранит или возьмет содержимое из безымянного регистра “”.

Кроме этих регистров общего назначения, в *Vim* также имеется несколько специальных. Самый полезный из них — регистр ‘0’. Он хранит только «выдернутый» текст, а безымянный регистр — также и удаленный текст.

Это полезно, когда вы хотите выдернуть текст, удалить несколько строк, а затем вставить его. Если вы просто использовали безымянный регистр, выдернутый текст заменится на удаленный. Если вы вставляете с “**0p**”, вы получите выдернутый текст, а не удаленный.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Модули расширения

Есть одна функция *Emacs*, связанная с выдергиванием, которой в *Vim* никогда не было, под названием «покойницкая [kill ring]». Она сохраняет все выдернутые, удаленные или измененные тексты и делает их доступными для каждого запущенного экземпляра *Emacs*. При каждом добавлении чего-то в «покойницкую» предыдущее содержимое остается, и пользователь может переключаться между всеми предыдущими «убитыми».

Vim, однако, имеет мощный скриптовый интерфейс, что означает возможность разнообразного его расширения и дополнения. Один пользователь создал скрипт *yankring*, который воссоздает «покойницкую» *Emacs*.

Все скрипты *Vim* можно найти в Интернете, на www.vim.org/scripts. Зайдя на этот сайт и поискав скрипт *yankring*, вы увидите ссылку для его скачивания.

Чтобы настроить *Vim* на добавление новых скриптов, вам первым делом нужно создать два новых каталога:

```
mkdir ~/.vim/plugins
mkdir ~/.vim/docs
```

После этого распакуйте архив и скопируйте содержимое его каталога модулей во вновь созданный каталог *.vim/plugins*, аналогично поступив с содержимым каталога *docs*. Сделав это, перезапустите *Vim*, обновите индекс подсказки по `:helptags $HOME/vim/doc`, и вы в дамках. Вот так устанавливаются все скрипты *Vim*.

Чтобы узнать, как использовать *Yankring*, запустите `:help yankring-tutorial`. Скриптов для *Vim* полным-полно.

Подстройка Vim «под себя»

Модули расширения – не единственный способ настроить *Vim*. *Vim* также поддерживает создание новых сокращений и раскладки клавиатуры, что позволяет придумать быстрые способы сделать почти все.

Начнем с сокращений. Здесь, в *LXF*, мы должны представить наш текст с подзаголовками, кодом и врезками по всему тексту. Наша разметка выглядит так:

```
/// Код ///
SomeFunction () {}
/// Код ///
```

Вводить повторяющиеся символы наподобие этих, особенно с надолго прижатой клавишей Shift, на редкость нудно! Чтобы помочь горю, можно просто создать новую аббревиатуру в *Vim*. Как подсказывает название, аббревиатура – это особая строка текста, которая при переходе в режим вставки заменяется на более длинный фрагмент текста. Добавление `iab codes` и `iab endc` к вашему

файлу *.vimrc*, например, означает, что если вы введете “codes”, затем пробел, оно будет заменено на `/// Код ///`.

Так гораздо проще, но будьте осторожны с теми строками, что вы используете для аббревиатур.

Аббревиатуры можно применять для замены типичных опечаток. Если вы часто печатаете ‘the’ как ‘teh’, задайте это как аббревиатуру, и опечатка всегда будет устранена.

Назначение клавиш

Назначение клавиш [key maps] отличается от аббревиатур. Это не предоставление быстрого способа ввести фиксированные текстовые строки: переназначение клавиш позволяет создавать альтернативные пути для выполнения различных действий в *Vim*.

Возьмите, к примеру, *Yankring*. Чтобы увидеть, что сейчас находится в *Yankring*, нужно выполнить `:YRShow`. Это не слишком эффективная команда, но вы можете ускорить процесс, повесив ее на одну клавишу.

В *Vim* есть два способа это сделать. Первый добавляет запись ‘map’ в файл *.vimrc*. Формат похож на формат записи ‘iab’: `map <F5>:YRShow<CR>` велит клавише **F5** запустить показ *Yankring*. Обратите внимание, что указанные нажатия клавиш именно те, что вы набрали бы, выполняя команду, в том числе `<CR>` (возврат каретки).

Такое применение `map` может, однако, вызвать проблемы, поскольку по умолчанию это рекурсивная команда. Это означает, что если вы назначите `j` для `gj`, а потом `d` для `j`, то обнаружите, что `d` будет выполнять не `j`, а `gj`. Чтобы обойти эту проблему, можно использовать команду `noremap`, которая создает отображения, не являющиеся рекурсивными. Вернувшись к нашему примеру с командой `noremap`, вы обнаружите, что `j` выполняет `gj`, а `d` выполняет `j`.

Это еще не все, люди

Хотя мы подошли к концу урока этого месяца, в *Vim* еще много чего осталось неохваченным. Обязательно прочтите отличные встроенные файлы помощи, особенно те, что составляют руководство пользователя, набрав `:help user`. По сравнению со справочной документацией это приятное чтение.

Самое главное, не переживайте, если не выучите все сразу. Для этого там просто слишком всего много. Вместо этого, изучайте по кускам, по мере надобности, и благодаря логике, лежащей в основе системы команд *Vim*, вы скоро обнаружите, что освоили сложный язык редактирования текста! **LXF**

«Переназначение клавиш — альтернатива выполнению действий.»

Придадим Vim элегантности

Аскеты среди нас могут думать, что внешнее оформление вашего текстового редактора не имеет значения: ну, видите вы слова на экране и можете делать работу, а все остальное только отвлекает. Отчасти это верно, но если вы целый день пялитесь в свой текстовый редактор, вам будет важен его приятный вид. Вам захочется, чтобы схема подсветки синтаксиса была осмысленной, а также контраста для удобства.

Если вы используете *gVim*, графический порт *Vim*, или ваш терминал поддерживает цвета (большинство современных термина-

лов это умеют), то можно настроить цвета *Vim* для отображения текста.

Цветовые схемы приходят как файлы *.vim*, и их надо сохранить в каталоге `~/.vim/colors`, чтобы пустить в дело. Сделав это, установите `:colorscheme <schemename>` в своем файле *.vimrc* для использования схемы.

Много разных схем есть на <http://tinyurl.com/7omvmuq>, и вы можете использовать ссылки в нижней части <http://code.google.com/p/vimcolorscheme-test/> для предпросмотра всех цветовых схем.



➤ Тест цветовой схемы Vim покажет, как выглядят цветовые схемы для различных типов кодов.

Arduino: Немного

В этой громкой статье маэстро **Ник Вейч** превращает свой Arduino в музыкальный инструмент, употребляя слово «настройка» в буквальном смысле.



Наш эксперт

Когда LXF только появился, его держали на плаву исключительно скрипты *Bash* от **Ника Вейча**. Потом их заменили «людьми», и это, по мнению Ника, стало шагом назад...

Во многих проектах Arduino – будь то хоровод светодиодов или контакты ЖК-экрана – ведущую роль играет свет. Это очень мило, но есть другие измерение, которым мы пока пренебрегали: это звук. Помогать перед кем-то светодиодами – штука хорошая, но люди ведь должны на них сначала посмотреть; прелесть же звука в том, что его не так легко игнорировать, в чем скоро убедятся ваши друзья и родные.

Создаем музыку

Простейший способ добыть звук – воспользоваться микросхемой тонального генератора. Они очень дешевы, требуют минимума дополнительных компонентов для подключения динамика, и ими легко управлять с помощью импульсов с одного из цифровых выходов Arduino. Такая схема слишком проста, чтобы описать ее здесь.

Сформировать собственный звук ненамного сложнее. Звук – это всего лишь ударная волна, и с подходящим генератором волн (например, динамиком) мы получим простые звуки прямо с выводов Arduino. Мало того, при всей несложности данной задачи, в ПО Arduino есть еще и специальная библиотека, которая об этом заботится.

Библиотека тонов формирует прямоугольные колебания (т.е. «вкл» и «выкл»), манипулируя внутренними генераторами тактов микросхемы Atmega. Она просто задает частоту и направляет результат на нужный вывод. Прямоугольные колебания дают грубый и сырой звук, но в этом есть и свои плюсы. В реальном мире вы вряд ли подключите эту схему к своей системе Hi-Fi, но она

прекрасно подойдет на роль дверного звонка. Помня обо всем этом, давайте покорежим классический опус – «Оду к радости» Бетховена. Так или иначе она нам подходит. Первое, что нам нужно знать – как сгенерировать звук. Это просто:

```
tone(pin, pitch, duration);
```

Обязательных аргументов у функции **tone** два: номер вывода (**pin**) и тон (**pitch**). Третий, необязательный аргумент – продолжительность звучания (**duration**). Номер вывода – обычный номер контакта Arduino. Тон – это произвольно указанная частота; воспроизводить стандартные ноты вы не обязаны.

Для вашего удобства, в библиотеке есть заголовочный файл с определениями нот, поэтому можно пользоваться такими обозначениями:

```
tone(7,1568);
```

```
tone(7,NOTE_G6);
```

Оба варианта вызова воспроизведут одну и ту же ноту [«соль» средней октавы, – прим. ред.]. Если продолжительность звучания (целое число миллисекунд на воспроизведение ноты) не указана, то в нужный момент придется его остановить:

```
noTone(pin);
```

В один момент времени можно воспроизводить несколько нот; в стандартном Duemilanove Arduino – до трех, но в большинстве случаев на практике есть ограничение в две ноты. Это связано с тем, что для получения прямоугольной волны используется встроенный генератор тактов, который также обеспечивает генерацию ШИМ-сигналов и функцию внутреннего таймера миллисекунд. Заняв его тремя нотами, вы больше не сможете отследить прохождение времени, что может стать проблемой. Учитывая качество звука, получаемого с такой схемой, часто достаточно вообще одного канала.

Сохраняем мелодии

Итак, звуки издавать мы теперь умеем, но как создать музыку? Для каждой ноты нам нужно знать три вещи – тон, время звучания и интервал до следующей ноты. В примерах из библиотеки мелодий эти параметры хранятся в различных массивах, что, с одной стороны, упрощает код, а с другой – усложняет его понимание. В идеальном случае всю информацию о мелодии нужно хранить в одном массиве. Тут нам помогут структуры – **struct**, это пользовательский тип данных, который легко определить и которым удобно пользоваться. Он пришел из стандартного языка C, на котором, кстати, мы и пишем большую часть кода для Arduino (удивлены? – а мы вроде об этом упоминали). Вот синтаксис объявления структуры:

```
struct mydata { <type> <propertyname>;  
<type> <propertyname>;  
...  
}
```

Структуру для ноты можно организовать так:

```
struct note{  
uint16_t pitch;
```

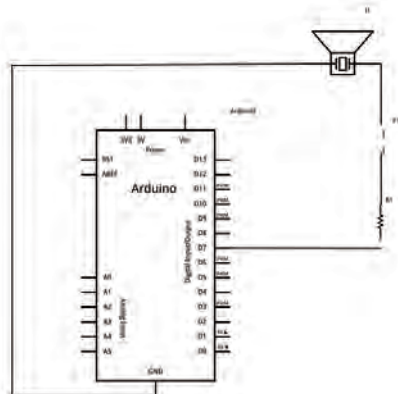
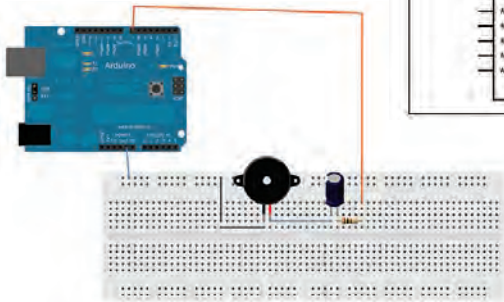
» **Месяц назад** Мы объединили два Arduino – пусть порезвятся вместе.



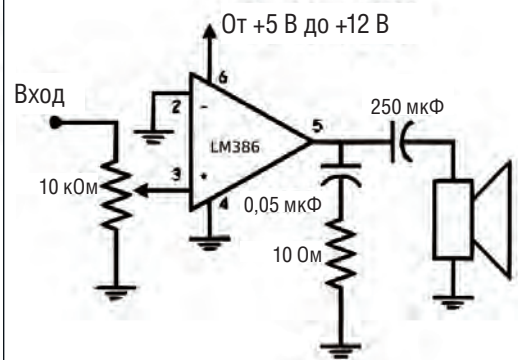
ПОШУМИМ

Идем дальше

Вывод можно подключить к готовому усилителю (но сначала подключите делитель напряжения! Линейные входы обычно используют напряжение 1 В, а не 5 В) или собрать свой собственный. Существует множество схем простых усилителей для больших динамиков, а поскольку качество звука будет не слишком хорошим, если не пропустить его через несколько стадий фильтрации, выбирайте легко узнаваемые мелодии.



► Рис. 1. Подключение динамика к Arduino. Рекомендуется, хотя и не обязательно, использовать конденсатор.



► Рис. 2. С помощью простого операционного усилителя/усилителя мощности можно подключить схему к более мощным и громким динамикам; либо подключите схему к отдельному усилителю через делитель напряжения.

```
uint8_t duration;
uint8_t pause;
};
note first={880,12,8};
int duration =first.duration*25;
```

Мы создали простую структуру данных с тремя значениями. Частота – 16-битное целое число (0–65535), а продолжительность звучания и пауза – 8-битные числа (0–255). Два последних значения являются скорее условными, чем истинными. Время нужно задавать в миллисекундах в диапазоне где-то от 50 до 2000. Но если хранить эти значения в 16-битных числах, для хранения каждой ноты понадобится на 50 % больше места (шесть байт вместо четырех), и мы сильно раздуем объем требуемой памяти. Так как нам вряд ли нужно воспроизводить ноты с миллисекундной точностью, проще хранить в памяти меньшие числа, а при использовании умножать их на коэффициент. Если умножать их на 25, получится диапазон от 0 до 6375, т. е. более шести секунд, и этого должно быть вполне достаточно.

Сами ноты тоже при желании можно уместить в один байт – если вы будете играть музыку, то скорее всего по нотам, а не звуками произвольной частоты, поэтому можно ограничиться диапазоном 255 нот (уж этого-то хватит!), который поместится в одном байте (еще одно беззнаковое 8-битное целое число). Наконец, стоит ввести обозначения для нот, которые мы хотим воспроизводить:

```
// notes in the melody:
enum { GG,A,B,C,D,E,F,G};
static const uint16_t frequency[] = {
    784, // GG
    880, // A
```

```
988, //B
1047,
1175,
1319,
1397,
1568//G
};
```

Объявление их как статических констант означает, что значения никогда не будут и не смогут измениться. Вы можете счесть это скверными попытками сэкономить используемое место, но иначе память растратится довольно быстро. Пусть наша мелодия не слишком длинная, но хранение данных таким образом сэкономит до 1 КБ даже на коротком наигрыше, и это важно, если вы намерены сделать в программе что-то еще.

Теперь можно перейти к мелодии. Для краткости мы приведем только первый такт «Оды к радости», но в листинге DVD приведена вся мелодия:

```
note tune[] = { (E,10,12),(E,10,12),(F,10,12) ,(G,10,12),
(G,10,12),(F,10,12),(E,10,12) ,(D,10,12),
(C,10,12),(C,10,12),(D,10,12) ,(E,10,12),
(E,12,14),(D,7,8),(D,24,32),
...
};
```

Пробелы при парсинге кода не воспринимаются, и вы можете отформатировать свой код так, чтобы его было проще читать или редактировать. Для каждой ноты приведены частота, продолжительность и пауза. Пауза всегда длиннее продолжительности, потому что в коде мы сделаем так, чтобы пауза включала время воспроизведения всей ноты. Наконец, нам нужен код для воспроизведения мелодии:

Что нам надо

- Небольшой динамик или пьезо-элемент.
- Любой Arduino.
- Также рекомендуется: конденсатор емкостью 250 мкФ и резистор сопротивлением 100 кОм.

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


```
void loop() {
for (int i = 0; i < (sizeof(tune)/3); i++) {
tone(7, frequency[tune[i].pitch], tune[i].duration*25);
delay(tune[i].pause*25);
}
delay(15000);
}
```

Этот код, разумеется, стоит обернуть в собственную функцию, а не повторять все время в главном цикле. Хотя в этих строках много чего происходит, код очевиден.

При инициализации цикла мы можем определить количество нот, получив размер массива (в байтах) и разделив его на три, так как для каждой ноты используется три байта.

Настраиваемся

В следующей строке мы воспроизводим ноту с помощью функции библиотеки тонов. Мы выдаем ее на вывод 7 (он не хуже любого другого), и это первый аргумент функции. Частоту ноты возьмем из созданного ранее массива. В качестве индекса мы используем значения из перечисления, которые хранятся в нашем массиве `tune` в поле структуры `pitch`. Так, `tune[i].pitch` содержит число от 0 до 7 (числами мы обозначили ноты, для простоты прочтения), которое используется в качестве индекса массива, где хранятся настоящие частоты.

Последний нужный нам параметр – продолжительность звучания. Он должен быть в миллисекундах, но мы для экономии памяти делили значения на 25, и теперь нужно умножить их на 25.

Пауза между нотами реализуется стандартной функцией `delay()`, которая аналогичным образом использует данные из массива `tune`. Чтобы гарантированно прервать воспроизведение, можно было вызвать функцию `noTone(7)`, но на практике в этом нет необходимости.

Для воспроизведения мелодии можно подключить небольшой динамик прямо к выводам Arduino. Но чтобы не повредить сам динамик, понадобится небольшой резистор – где-то в диапазоне от 100 Ом до 1 кОм. Если вы используете пьезоэлемент или пьезодинамик, без резистора можно и обойтись: они часто обладают большим внутренним сопротивлением, рассчитанным на сигналы логических уровней. Формула, которую вам нужно знать, выглядит так:

$$P = \frac{V_{\text{peak}}^2}{2R}$$

Это всего лишь приближение! Динамики устроены сложнее, чем вы думали – они содержат индуктивную нагрузку, и в определении мощности по отношению к динамике есть масса тонкостей. Нам же важно просто его не сжечь, поэтому достаточно и приближенной формулы; но требуются меры предосторожности. В данном случае R – резистивная нагрузка (сопротивление динамика должно быть указано на его задней поверхности – обычно это 4, 8 или 16 Ом), а V – пиковое напряжение – в нашем случае 5 В без использования усилителя мощности. С нагрузкой 16 Ом это даст нам примерно 1,5 Вт.

Предотвращаем перегрузку

Что произойдет при перегрузке динамика? Пьезоэлемент, если не доходить до беспредела, просто будет издавать ужасные звуки, пока не выйдет из строя, но вообще-то на удивление вынослив. В обычном динамике с бумажным конусом и сердечником вы либо повредите сердечник, либо расстрясите его на части, либо повредите конус – а часто и то, и другое; поэтому здесь тоже нужно принять меры предосторожности. Из-за способа своего изготовления динамики чреваты также опасностью для Arduino.

Когда зуммер — не зуммер

Пьезозуммер – стандартный пьезоэлемент, который ведет себя очень похоже на динамик. Пьезокристаллы совершают колебания при воздействии тока и генерируют ток под воздействием колебаний, поэтому из них получаются прекрасные преобразователи (их также можно использовать как датчики

давления). Раньше зуммеры содержали какой-нибудь осциллятор для генерации собственных сигналов. Теперь, даже несмотря на то, что иногда их называют зуммерами, многие пьезоэлементы представляют собой просто замену динамикам (т.е. вам нужно подать на них свой сигнал), хотя

по своим звуковым характеристикам они больше подходят для воспроизведения «воюющих» звуков, а не музыки – и это одна из причин, по которой аудиооткрытки так бесят именинников! Сбравшись приобрести такую, предвзвременно озаботьтесь прослушиванием покупки...



» В онлайн-руководстве по Arduino есть прекрасное описание побитовых функций: <http://www.arduino.cc/en/Reference/Bitwise>.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Крошечный пьезоэлемент не причинит ему вреда, но чем больше индуктивность, тем больше шумы скачков напряжения и тем больше мощность цепи. Хотя в цепях для управления динамиками с помощью Arduino, которые вам могут встретиться, его обычно нет, рекомендуется добавить развязывающий конденсатор емкостью около 250 мкФ, подключив его выход к «плюсу» динамика. Он не только сглаживает переходные процессы, но и поглощает постоянную компоненту сигнала, и сердечник будет перемещаться в обоих направлениях, как и предписано его природой.

Дополнительный звук

Мы вступаем на территорию аудиоцепей, и она может показаться вам незнакомой, потому что в аудиоцепях присутствуют циклические напряжения, которые роднят их с цепями переменного, а не постоянного тока – поэтому мы и добавили развязывающий конденсатор в нашу главную схему.

Дело в том, что в цепях переменного тока компоненты ведут себя по-другому, и это заметнее всего на примере скромного конденсатора. В цепях постоянного тока мы пользовались конденсаторами для буферизации напряжения и сглаживания переходных процессов, и, возможно, иногда использовали их в качестве примитивного таймера. С переменным током они ведут себя иначе: они его пропускают. На самом деле никакого тока через них не течет – они просто сохраняют и освобождают электрическую энергию (это если вы не пробьете конденсатор, подав на него слишком большое напряжение, чего мы делать не советуем, потому что он щелкнет и премеерзко завоняет), но из-за циклической зарядки и разрядки обоих пластин создается *видимость* прохождения тока.

Это поведение зависит только от частоты приложенного напряжения и, разумеется, емкости самого конденсатора – если вникнуть, то чем выше частота, тем больше циклов зарядки и разрядки выполняется каждую секунду, и тем больше ток, «проходящий» через конденсатор. Свойство зависимости от частоты называется реактивным сопротивлением компонента и определяется по формуле

$$\text{Reactance}(X) = \frac{1}{2\omega fC}$$

Мелодии удлиняются

Хотя в нашей мелодии каждая нота описана тремя байтами, можно легко уместить все три значения высоты ноты, продолжительности звучания и паузы в один байт. В одной мелодии вряд ли будет задействовано более трех октав – это 32 значения. Как видно из нашего примера, продолжительность звучания и пауза также не пре-

вышали 32 каждая. Сложив все это вместе, мы получим 96 – меньше полбайта данных. Составление такой мелодии может потребовать больше усилий, но извлечение информации с помощью битовых масок – не слишком сложная и не слишком долгая задача для процессора, поэтому такая возможность всегда есть.

Это сопротивление в чем-то аналогично обычному сопротивлению резистора в цепях постоянного тока. Но так как оно зависит от частоты проходящего через компонент сигнала, то чрезвычайно удобно для создания аудиофильтров.

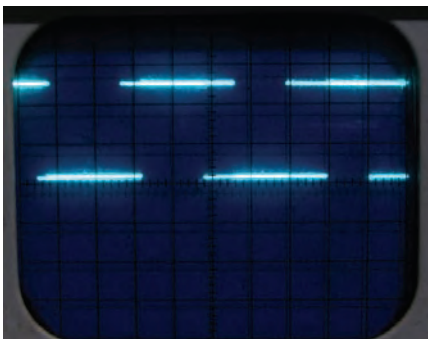
Простой резистор и конденсатор (RC) можно использовать в качестве делителя напряжения сигнала, изменяющего его уровень в зависимости от частоты. Это означает, что сигналы с одними частотами пройдут через эту цепочку без изменений, а сигналы с другими – существенно снизятся; что и пригодится для подавления нежелательных сигналов и наводок в нашей системе. Частотный диапазон звука, слышимого человеческим ухом, нам известен, и можно отфильтровать ненужные колебания, подобрав подходящий конденсатор.

В зависимости от резистора и конденсатора можно создать фильтр верхних частот (т.е. отрезать нижние частоты – таким способом удобно избавляться от шума электросети питания) или фильтр нижних частот (с его помощью удобно отфильтровывать другие помехи, такие как шум от генератора тактов), или объединить два фильтра и получить полосовой фильтр.

Мы не будем слишком углубляться в теорию звука, потому что это не тема нашей серии, да и вычисления очень усложняются – но мы и не собираемся ею пренебрегать. Присоединяйтесь к нам в следующий раз для увлекательного путешествия с генерацией синусоидальной волны со сдвигом частоты для эмуляции терменвокса. **LXF**

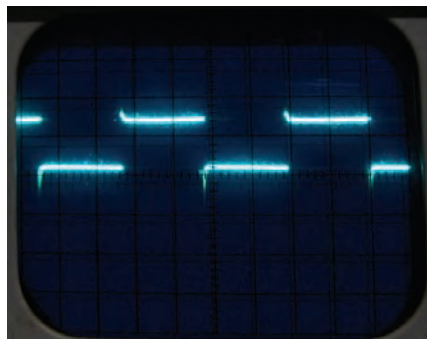
Сигналы с простой схемы

Осциллографы есть не у всех, поэтому вот примеры выходных сигналов нашей схемы...



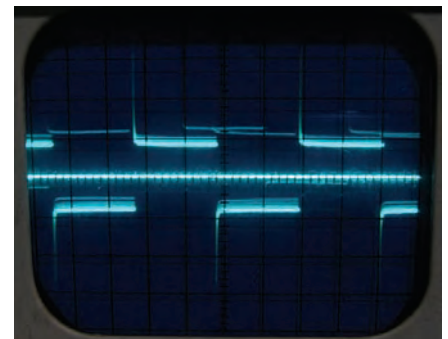
1 Сигнал 1

Выход Arduino, к которому ничего не подключено. Нулевая точка – в середине, и вы видите аккуратные прямые линии и четкие переключения.



2 Сигнал 2

С динамиком, подключенным через низкоомный резистор, сигнал приобретает скачки в точках переключения – они вызваны сердечниками динамика, но не особенно опасны при маленьком динамике.



3 Сигнал 3

С развязанным выходом низкие значения опускаются ниже центральной линии. Емкость также несколько искажает форму сигнала, но не слишком ее портит.

» **Через месяц** Перепрограммируем генератор тактов на другие волны.



Erlang: Изучим

Андрей Ушаков увлечен взаимодействием между процессами.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

Современный мир разработки ПО сложно представить без такого средства, как многозадачность. Многозадачность позволяет решать многие задачи за приемлемое время благодаря тому, что многие части одной задачи выполняются одновременно. Многозадачность – это одна из самых сложных областей программирования. Язык Erlang в свое время был специально разработан для решения большинства технических проблем, связанных с многозадачностью. Эта статья открывает большой цикл, посвященный созданию многозадачных и распределенных приложений средствами языка Erlang.

А начнем мы наш разговор с определения основных терминов, связанных с темой нашего разговора. Первым делом дадим определение термину «многозадачность». Многозадачность – это свойство языка программирования, среды выполнения и операционной системы по обеспечению одновременного (или псевдоодновременного) выполнения нескольких задач. Если язык программирования не является многозадачным, то и беспокоиться нам не о чем: все задачи у нас выполняются последовательно. Если среда выполнения или операционная система не являются многозадачными, то максимум, что мы сможем получить – это иллюзию многозадачности. Эта иллюзия заключается в том, что поток выполнения у нас в реальности всего один; среда выполнения или ОС переключают его на выполнение разных фрагментов кода.

Если и язык программирования, и среда выполнения, и ОС поддерживают многозадачность, то чтобы добиться одновременного выполнения нескольких задач, необходимо выполнить два условия. Во-первых, истинную многозадачность мы можем получить, запуская несколько задач на многопроцессорном или многоядерном компьютере, или на нескольких компьютерах, соединенных сетью. Во-вторых, и это, пожалуй, самое важное, для одновременного выполнения нескольких задач в рамках одной программы программа изначально должна разрабатываться как многозадачная.

Многозадачность бывает двух типов: основанная на процессах и основанная на потоках. Говоря о многозадачности, основанной на потоках, мы имеем в виду, что у нас есть несколько потоков выполнения кода в рамках одного процесса. При такой конфигурации несколько задач одновременно выполняются в одном адресном пространстве; это упрощает взаимодействие между задачами, но при неправильной реализации данного взаимодействия может привести к несогласованности и порче данных.

Многозадачность, основанная на процессах, подразумевает, что у нас есть несколько процессов на одном или нескольких компьютерах. При такой конфигурации у каждой задачи свое адресное пространство, что предотвращает многие случаи возможной порчи состояния одной задачи другой задачей, но усложняет взаимодействие между задачами. В дальнейшем мы будем различать ситуации, когда несколько процессов выполняются на одном компьютере и когда несколько процессов выполняются на разных компьютерах, соединенных сетью. Связано это с тем, что способы взаимодействия между процессами зависят от того, на одном или на разных компьютерах процессы выполняются.

Многозадачность, основанную на процессах, выполняющихся на разных компьютерах, мы будем рассматривать отдельно. Такой вариант многозадачности называют распределенной системой

(а саму такую многозадачность – распределенными вычислениями). Главное преимущество распределенных систем перед нераспределенными в том, что их мощность (количество компьютеров, объединенных сетью) можно наращивать бесконечно. Минусы подобных систем в том, что их достаточно сложно создавать и поддерживать их работу (вполне очевидно, что чем больше такая распределенная система, тем это делать сложнее). Кроме того, далеко не всякую задачу имеет смысл решать в распределенной системе; но это тема для отдельной статьи.

Вместо этого давайте рассмотрим класс задач, решение которых в распределенной системе дает существенный выигрыш по сравнению с решением нераспределенным и однозадачным способом. Это задачи по поиску элемента из некоторого множества методом полного перебора (или методом «грубой силы», англ. brute force): мы перебираем все элементы из множества, пока не найдем удовлетворяющий нас элемент.

Очевидно, что если множество элементов упорядочено и существует связь между критерием поиска (близостью элемента к искомому) и порядком элементов, то метод поиска полным перебором в такой ситуации не оптимален. Также вполне очевидно, что если мы разделим множество элементов на N равных подмножеств и начнем поиск в этих подмножествах одновременно, то среднее время поиска будет в N раз меньше среднего времени поиска во всем множестве элементов. Поэтому вполне логично, что такой класс задач хорошо подходит для решения на распределенных системах. К такому классу задач относится, например, задача подбора пароля по хэшу MD5 (хотя для этой задачи существует и альтернативный подход к решению, с использованием радужных таблиц).

Пойдем дальше и остановимся на еще одном понятии, без которого невозможно создавать более-менее сложные многозадачные программы: это взаимодействие между задачами (обычно называемое взаимодействием между процессами или IPC). Средства, применяемые для такого взаимодействия, зависят от вида используемой многозадачности (основанной на потоках, основанной на процессах на одном компьютере или основанной на использовании распределенных вычислений), а также от языка программирования (если такие средства встроены в язык), среды выполнения и ОС. Вполне логично, что средства, применяемые для взаимодействия процессов в распределенной системе, могут быть применены и для взаимодействия процессов на одном компьютере, и для взаимодействия потоков в одном процессе. А средства, применяемые для взаимодействия процессов на одном компьютере, пригодны и для взаимодействия потоков внутри процесса.

Возникает вопрос: зачем нам использовать более общие средства взаимодействия в случае, когда можно применить более подходящие для данного типа многозадачности? Одна из причин такого подхода состоит в том, что, разрабатывая изначально взаимодействие между задачами таким способом, мы получаем возможность использовать наш код для любых видов многозадачности. Кроме того, при таком подходе мы можем избежать или минимизировать возникновение одновременного доступа к данным (точнее говоря, нам надо избегать одновременного доступа

МНОГОЗАДАЧНОСТЬ

к данным, если мы хотим изменять эти данные) и, как следствие, возможного повреждения и несогласованности данных.

Теперь давайте разберемся, какие есть средства для создания многозадачных программ. В качестве примера такого средства рассмотрим ОС Linux. Первый вопрос, который встает перед нами – как создавать новые задачи. Начнем с создания потоков. В Linux для работы с потоками у нас есть библиотека *pthread* (интересно, что потоки в Linux – это процессы, разделяющие ресурсы с процессом, который их создал); для создания новых потоков используется функция *pthread_create*. Перейдем к созданию процессов. Для создания процессов у нас есть следующие библиотечные функции: функция *fork* для создания нового процесса, семейство функций *exec* для запуска в рамках процесса другой программы, функция *system* для выполнения команд. Следует сказать, что создавать новые процессы мы можем только на локальном компьютере; возможности создать новый процесс (и запустить в нем какую-либо программу) у нас нет. Поэтому при построении распределенной системы необходимо предусмотреть автоматический старт (при старте системы) некоторого процесса для взаимодействия узлов этой распределенной системы.

Пришла пора посмотреть, какие есть средства взаимодействия между задачами в ОС Linux. Начнем с многозадачности, основанной на потоках; для взаимодействия задач у нас есть мьютексы, блокировки чтения-записи, условные переменные, возможность ожидания завершения потока. Все эти средства основаны на использовании блокировок потока выполнения. Множество средств для многозадачности, основанной на локальных процессах, достаточно разнообразно: именованные и неименованные каналы, файлы, очереди сообщений, семафоры, разделяемая память, сокеты домена Unix, возможность ожидания завершения процесса. Если для взаимодействия процессов мы используем разделяемую память или файлы, то при неаккуратной работе с данными (например, когда несколько процессов одновременно пытаются эти данные изменить) данные могут испортиться; в этом случае необходимо использовать дополнительные средства синхронизации, такие как блокировка записей в файле, семафоры, обмен сообщениями. И, наконец, для многозадачности в распределенной среде у нас есть лишь одно средство взаимодействия между задачами – сокеты (правда, конфигурация сокетов может быть разнообразной – например, сокеты с установлением соединения и без).

После общего обзора, что такое многозадачность, и обзора средств для создания многозадачных приложений на примере конкретной платформы мы можем перейти непосредственно к основной теме статьи – многозадачности в языке Erlang. Язык Erlang изначально был разработан для создания многозадачных и распределенных приложений, поэтому средства для создания задач и взаимодействия между ними имеют поддержку на уровне языка (в чем очень скоро мы убедимся).

В отличие от большинства языков и платформ, в языке Erlang существует только один тип задач, называемый процессами. Задачи в языке Erlang не являются истинными процессами, т. е. не обладают собственным адресным пространством. Это означает, что несколько процессов могут выполняться в пределах одного

Типы псевдопараллельной многозадачности

» Невытесняющая многозадачность

Тип многозадачности, при котором операционная система может загрузить более одного приложения в память, но время процессора предоставляется только одному из них – основному приложению. Остальные приложения являются фоновыми. Для предоставления процессорного времени фоновому приложению его необходимо активировать. Подобная многозадачность может быть реализована не только в ОС, но и с помощью программ – переключателей задач.

» Кооперативная многозадачность Тип многозадачности, при котором следующая задача выполняется только после того, как текущая задача явно провозгласит себя готовой отдать процессорное время другим задачам. Как частный случай, такое объявление подразумевается при попытке захвата уже занятого объекта блокировки, а также при ожидании поступления следующего сообщения от подсистемы пользовательского интерфейса.

» Вытесняющая многозадачность Вид многозадачности, в котором ОС сама

передает управление от одной выполняемой программы другой в случае завершения операций ввода-вывода, возникновения событий в аппаратуре компьютера, истечения таймеров и квантов времени или же поступлений тех или иных сигналов от одной программы к другой. В этом виде многозадачности процессор может быть переключен с исполнения одной программы на исполнение другой без всякого пожелания первой программы и буквально между любыми двумя инструкциями в ее коде. Распределение процессорного времени осуществляется планировщиком процессов. К тому же каждой задаче может быть назначен пользователем или самой операционной системой определенный приоритет, что обеспечивает гибкое управление распределением процессорного времени между задачами (например, можно снизить приоритет ресурсоемкой программе, снизив тем самым скорость ее работы, но повысив производительность фоновых процессов). При этом обеспечивается более быстрый отклик на действия пользователя.

экземпляра среды выполнения Erlang (на одной виртуальной машине Erlang). С другой стороны – и это роднит процессы Erlang с истинными процессами – процессы Erlang изолированы друг от друга (в отличие от, например, потоков в случае многозадачности, основанной на потоках). Изолированность процессов друг от друга заключается в следующих аспектах. Во-первых, данные одного процесса не доступны никому, если этого не пожелал сам процесс. Во-вторых, если во время работы возникает необработываемое исключение, то процесс Erlang будет завершен (так же как и некоторые связанные с этим процессом процессы; но об этом – на одном из следующих уроков). При этом завершение сбойного процесса Erlang не коснется других (не связанных с ним) процессов, вне зависимости от того, выполняются ли они в этом же экземпляре среды выполнения Erlang или нет. Такое поведение отличается от поведения при возникновении необработываемого исключения в одном из потоков выполнения какого-либо процесса: в этом случае обычно завершается процесс целиком, в том числе и другие потоки выполнения.

Процессы Erlang легковесны, их создание и завершение достаточно дешево (по сравнению, например, с созданием процессов и потоков в Linux), поэтому вопрос о том, использовать или нет многозадачность при решении той или иной задачи, должен

»

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

Проблемы в многозадачных средах

» **Голодание [starvation]** Задержка времени от пробуждения потока до его вызова на процессор, в течение которой он находится в списке потоков, готовых к исполнению. Возникает по причине присутствия потоков с большими или равными приоритетами, которые исполняются все это время. Негативный эффект заключается в том, что возникает задержка времени от пробуждения потока до исполнения им следующей важной операции, что задерживает исполнение этой операции, а следом за ней и работу многих других компонентов. Голодание создает узкое место в системе и не дает выжать из нее максимальную производительность, ограничиваемую только аппаратно обусловленными узкими местами.

» **Гонки [race condition]** Недетерминированный порядок исполнения двух путей

кода, работающих с одними и теми же данными и исполняемыми в двух различных задачах. Приводит к зависимости порядка и правильности исполнения от случайных факторов.

» **Инверсия приоритета** Пусть поток L имеет низкий приоритет, поток M – средний, поток N – высокий. Пусть поток L захватил объект блокировки (например, мьютекс) и, выполняясь с удержанием объекта блокировки, прерывается пробудившимся по какой-то причине потоком M, который имеет более высокий приоритет. Пусть поток N также пытается захватить этот объект блокировки. В такой ситуации поток N ждет завершения текущей работы потоком M, т.к., пока поток M исполняется, низкоприоритетный поток L не получает управления и не может освободить захваченный объект блокировки.

решаться только исходя из факта, можно ли распараллелить алгоритм задачи или нет. Конечно, следует сказать, что возможны ситуации, когда применение многозадачного решения будет неоправданно (когда, например, необходимо нескольким задачам работать одновременно с большим набором данных, расположенных в памяти), но этому мы посвятим один из последующих уроков.

Как уже говорилось, в языке Erlang у нас есть всего один тип задач на все случаи жизни – процессы Erlang. Мы уже поняли, что при желании использовать многозадачность в рамках одного экземпляра среды выполнения Erlang достаточно создать необходимое количество процессов Erlang. Но что делать, если мы хотим создать процессы Erlang на нескольких экземплярах среды выполнения Erlang – как на локальной машине, так и в случае распределенной системы? Для понимания этого, давайте введем понятие узла: узел – это именованный экземпляр среды выполнения Erlang. Для создания узла достаточно при ее запуске задать короткое (при помощи ключа `-sname`) или длинное (при помощи ключа `-name`) имя узла (этими ключами и разницей между ними мы более подробно займемся на одном из следующих уроков). После того, как узел с определенным именем создан, мы можем создать на нем необходимое нам количество процессов с некоторого другого узла. При этом создание процесса на каком-либо узле практически ничем не отличается от создания процесса в локальной среде выполнения Erlang: для создания процесса на узле нам необходимо лишь указать его имя, дополнив рядом параметров.

Может показаться, что такая возможность удаленного создания процессов – это большая дыра в системе безопасности. На самом деле это не так: средствами среды выполнения Erlang мы можем управлять тем, кто и какой код может запускать на удаленном узле, но об этом мы поговорим на одном из следующих уроков.

Для понимания того, как создавать многозадачные программы в языке Erlang, нам осталось поговорить еще на одну тему – взаимодействие между задачами. Взаимодействие между задачами в языке Erlang осуществляется при помощи обмена сообщениями. Процесс может послать любому известному ему процессу любое сообщение – любой объект языка Erlang. Что означает, что один процесс знает о другом процессе? Это означает, что у процесса есть идентификатор (называемый **Pid**) другого процесса, либо

другой процесс зарегистрировал свое имя, которое знает данный процесс. Идентификатор другого процесса мы можем знать, если мы сами его создали либо этот другой процесс послал нам сообщение, содержащее его идентификатор (каждый процесс может легко узнать свой идентификатор). Сообщение от одного процесса другому посылается асинхронно, т.е. мы не блокируем процесс-отправитель, пока это сообщение не будет доставлено процессу-получателю. Посылка сообщения от одного процесса другому процессу является надежной; это означает, что среда выполнения Erlang гарантирует доставку сообщения получателю.

Давайте посмотрим, как у нас обстоят дела со стороны получения сообщений. Каждый процесс в языке Erlang имеет связанную с ним очередь сообщений. Как только сообщение доставляется средой выполнения Erlang до процесса, это сообщение размещается в конце очереди сообщений данного процесса. Когда процесс пытается получить сообщение, удовлетворяющее некоторым критериям, то он просматривает последовательно свою очередь сообщений и извлекает из нее первое удовлетворяющее критериям отбора сообщение. Если удовлетворяющего критериям отбора сообщения нет, то процесс, инициировавший получение, переходит в состояние ожидания. Это ожидание длится до тех пор, пока среда выполнения Erlang не доставит процессу какое-либо новое сообщение. Как только новое сообщение будет доставлено, процесс поиска сообщения, удовлетворяющего критериям отбора, будет запущен заново. Понятно, что процесс может бесконечно ожидать удовлетворяющее его критериям сообщение; однако мы можем повлиять на максимальное время ожидания, задав его в выражении получения сообщения. Обмен сообщениями между процессами, несмотря на унификацию многозадачности в языке Erlang, не единственное средство для взаимодействия между задачами. Так как, помимо общения друг с другом, задачи предназначены и для взаимодействия с внешним миром, то в наши руки попадают такие средства, как сокеты, файлы и т.д.

После обсуждения принципов реализации многозадачности в языке Erlang пришла пора посмотреть на конкретные средства ее реализации. Для создания новых процессов у нас есть целое семейство BIF: **spawn/1**, **spawn/2**, **spawn/3**, **spawn/4**. Функции **spawn/1** и **spawn/3** предназначены для создания новых процессов на локальном экземпляре среды выполнения, функции **spawn/2** и **spawn/4** – на удаленном узле. Когда мы создаем новый процесс (одной из этих функций), мы должны указать задачу, которую этот процесс будет выполнять. Задачей всегда является некоторая функция. Функции **spawn/1** и **spawn/2** задают выполняемую задачу в виде ссылки на произвольную функцию. Функции **spawn/3** и **spawn/4** задают выполняемую задачу в виде **MFA**, где **M** – это модуль, **F** – некоторая экспортируемая из модуля **M** функция, **A** – список аргументов, передаваемых данной функции. Разница между этими подходами при задании выполняемой задачи состоит в том, что при помощи ссылки на функцию мы можем задать как экспортируемую функцию, так и анонимную функцию и не экспортируемую функцию из текущего модуля; при помощи подхода **MFA** мы всегда задаем экспортируемую функцию.

Для создания новых процессов у нас также имеется семейство BIF **spawn_link**. Разница между этим семейством функций и семейством функций **spawn** в том, что семейство функций **spawn_link** не только создает новый процесс, но и создает связь между новым процессом и текущим (разговор о связи между процессами тоже ждет нас в будущем). Все функции обоих семейств возвращают идентификатор процесса или **Pid**.

Для отправки сообщения от одного процесса другому процессу мы используем выражение **Process! Message**, где **Process** – выражение, определяющий целевой процесс (значением этого

» **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.

выражения является либо идентификатор процесса, либо имя зарегистрированного процесса), а **Message** – отправляемое сообщение (любой объект языка Erlang). Результатом этого выражения отправки сообщения является значение самого отправляемого сообщения **Message**. Помимо этого выражения отправки сообщений, существует еще целое семейство функций, определенных в модуле **erlang** для данной задачи. К этому семейству относятся функции **erlang:send/2**, **erlang:send/3**, **erlang:send_after/3**, **erlang:send_nosuspend/2**, **erlang:send_nosuspend/3**. Эти функции (за исключением функции **erlang:send/2**, которая идентична выражению отправки сообщения) позволяют настроить процесс отправки сообщений. Для решения задачи отправки сообщения у нас есть несколько возможных вариантов, а для получения сообщения только один – выражение **receive**. Выражение для получения сообщений **receive** имеет следующий вид:

```
receive
  Pattern1 [when GuardSeq1] -> Body1;
  ...;
  PatternN [when GuardSeqN] -> BodyN
[after ExprT -> BodyT]
end
```

В этом выражении получения сообщения каждое подвыражение вида **Pattern_i [when GuardSeq_i]** определяет критерий, по которому то или иное сообщение из очереди сообщений процесса может быть выбрано для обработки. В подвыражении **Pattern_i [when GuardSeq_i]** часть **Pattern_i** является выражением соответствия шаблону [pattern-matching], а часть **when GuardSeq_i** является выражением охраны. Если будет найдено сообщение, удовлетворяющее одному из критериев **Pattern_i [when GuardSeq_i]**, то значением выражения **receive** будет подвыражение **Body_i**, соответствующее подвыражению критерия выбора. Выражение **after ExprT -> BodyT** задает поведение, если в очереди сообщений не оказалось подходящего сообщения: в этом случае выполнение процесса приостанавливается максимум на **ExprT** миллисекунд. Если в течение **ExprT** миллисекунд в очереди сообщений процесса не появится сообщения, удовлетворяющего перечисленным выше критериям, то процесс закончит ожидание подходящего сообщения, и значением выражения **receive** будет подвыражение **BodyT**. Значением подвыражения **ExprT** также может быть атом **infinity**; в этом случае процесс будет ждать сообщения, удовлетворяющего одному из критериев, бесконечно долго. Выражение **after ExprT -> BodyT** не является обязательным; если оно отсутствует в выражении **receive**, то это эквивалентно случаю бесконечного ожидания подходящего сообщения процессом.

А теперь пришла пора практики: давайте на простом примере посмотрим, как работать с многозадачностью в языке Erlang. В этом примере мы из основного процесса создадим рабочий процесс и будем взаимодействовать с ним при помощи сообщений. Давайте рассмотрим сначала функцию (выполняемую задачу) рабочего процесса **example_worker/1**:

```
example_worker(Master) ->
  receive
    {ping, From } ->
      io:format("ping from master ~n"),
      From ! {pong, self()},
      example_worker(Master);
    _Other ->
      io:format("unknown message ~n"),
      example_worker(Master)
  after 1000 ->
    io:format("timeout ~n"),
    erlang:send(Master, timeout)
  end.
```

Эта функция содержит выражение для получения сообщений **receive**. Выражение **receive** умеет получать сообщения вида **{ping,**

From}, где **From** – любой объект языка Erlang (мы надеемся, что **From** – это идентификатор процесса отправителя, но при необходимости это можно проверить явно при помощи **BIF is_pid/1**) и все остальные сообщения. Обычно все остальные сообщения (т.е. сообщения, не подпадающие ни под один другой критерий) получают для того, чтобы эти «мусорные» сообщения не накапливались в очереди сообщения процесса. Кроме того, выражение **receive** содержит секцию **after**, которая задает максимальное время ожидания в 1000 миллисекунд.

При получении любого сообщения мы выводим информацию на экран и, при помощи хвостовой рекурсии, закликаем выполнение функции **example_worker/1**. Таким образом, функция **example_worker/1** является функцией обработки сообщений. Кроме того, при получении сообщения **{ping, From}** мы отсылаем процессу, пославшему нам это сообщение, сообщение **{pong, self()}**. **BIF self/0** возвращает идентификатор текущего процесса, но у нас нет никакого способа узнать идентификатор процесса, пославшего нам сообщение, кроме как включить этот идентификатор в само сообщение. Именно этого мы ожидаем при получении сообщения **{ping, From}**, и именно это мы делаем, когда отсылаем сообщение **{pong, self()}**.

И, наконец, при наступлении таймаута в выражении **receive** мы отсылаем сообщение **timeout** главному процессу при помощи функции **erlang:send/2** (для демонстрации такой возможности) и обрываем цикл обработки сообщений.

Теперь давайте взглянем на функцию главного процесса **example/0**:

```
example() ->
  Master = self(),
  Worker = spawn(fun() -> example_worker(Master) end),
  Worker ! hello,
  Worker ! {ping, Master},
  Worker ! hi,
  receive
    timeout -> io:format("timeout in worker ~n")
  end,
  receive
    {pong, Worker} -> io:format("pong from worker ~n")
  end.
```

В этой функции мы создаем рабочий процесс (при помощи **BIF spawn/1**, которая возвращает идентификатор созданного процесса), после чего посылаем рабочему процессу ряд сообщений. Среди этих сообщений находится как известное рабочему процессу сообщение **{ping, Master}**, так и неизвестные **hello** и **hi**. После этого, при помощи выражений **receive**, мы принимаем сообщения от рабочего потока. Заметьте, что принимать сообщения мы можем в любом порядке, а не в том, в котором они были посланы. Если мы запустим на выполнение функцию **example/0** (создав модуль с экспортируемой функцией **example/0**), то мы получим следующий вывод в консоли среды выполнения Erlang:

```
unknown message
ping from master
unknown message
timeout
timeout from worker
pong from worker
```

Мы сегодня сделали первый, но очень важный шаг в сторону многозадачности в языке Erlang. Написание многозадачных программ – достаточно сложная область, как в плане алгоритмов, так и в плане технических средств. Язык Erlang позволяет снять сложность технического плана и позволяет нам сосредоточиться только на соответствующих многозадачных алгоритмах. Мы это увидели сегодня (пускай это было не очень очевидно), мы это увидим еще не раз. А на следующем уроке мы поговорим о средствах для создания устойчивых к ошибкам программ. **LXF**



Применим VPN

Хотите дать доступ к вашей сети законным пользователям, отогнав интернетовский сброд? **Нейл Ботвик** учит это делать с OpenVPN.



Наш эксперт

Нейл Ботвик
У Неила стоит по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не выдаст вам местоположение своего центрального сервера.

Среди пользователей Linux популярны вопросы типа «Я хочу сделать службу XYZ доступны через Интернет, но только для авторизованных пользователей. Как это сделать безопасно?», где XYZ может быть VNC, NFS, Samba или локальный web-сервер. Ответ зависит от конкретного сервиса, который вы намерены предлагать, и от методов, предусмотренных сервисом для аутентификации и безопасности. Одни лучше, другие хуже, но это все равно потенциально означает вашу обязанность заботиться о нескольких сервисах, обеспечивая безопасность всех. С вашей локальной сетью таких проблем нет, потому что у вас только зарегистрированные пользователи – разве что в ваши компьютеры может забраться кто попало.

Альтернатива оголению отдельных частей сети для доступа из Интернета – разрешить авторизованным пользователям подключиться к сети извне так, как если бы они были подключены к локальной сети напрямую. Virtual Private Network (VPN) как раз и устанавливает безопасный, зашифрованный туннель между компьютером и сетью. Хотя туннель проходит через публичное интернет-соединение, весь трафик между вашим компьютером и сетью остается внутри туннеля, зашифрованный и защищенный.

Есть несколько различных реализаций VPN, но мы здесь воспользуемся OpenVPN. Это, как следует из названия, полностью открытая система, и она использует для соединения солидный *OpenSSL*. Прежде всего настройте маршрутизатор на проброс порта 1194 на компьютер, предназначенный на роль сервера *OpenVPN* в вашей локальной сети, иначе подключиться вы не сможете. В простейшем случае, можно подключить компьютер к сети, выполнив первую команду на компьютере в сети (сервере *OpenVPN*), а вторую команду – на удаленном компьютере, откуда вы хотите подключиться к сети (клиенте):

```
sudo openvpn --remote адрес_клиента --dev tun1 --ifconfig 10.0.1.1 10.0.1.2
sudo openvpn --remote адрес_сервера --dev tun1 --ifconfig 10.0.1.2 10.0.1.1
```

Опция **--remote** дает имя или IP-адрес компьютера, к которому подключаются, а два адреса, выданных **--ifconfig**, выделяются на локальном и удаленных компьютерах, соответственно – именно поэтому во второй команде они идут в обратном порядке. Эти

HTTP	443	Both	192.168.1.7	443
FTP	21	Both	192.168.1.7	21
IMAP	143	Both	192.168.1.7	143
IMAPS	993	Both	192.168.1.7	993
POP3	110	Both	192.168.1.7	110
POP3S	995	Both	192.168.1.7	995
Mythweb	8080	Both	192.168.1.9	80
SVN	3690	Both	192.168.1.7	3690
OpenVPN	1194	Both	192.168.1.3	1194
Monit	2812	Both	192.168.1.9	2812
Motion	8448	Both	192.168.1.1	8448
	0	Both	0.0.0.0	0
	0	Both	0.0.0.0	0

► Не забудьте установить маршрутизатор на переадресацию запросов, поступающих на порт 1194, на тот же номер порта на компьютере с сервером *OpenVPN*.

адреса должны лежать в одном из зарезервированных частных диапазонов, но в другой подсети вашей локальной сети, не то произойдут всякие ужасы. VPN – это отдельная сеть, а сервер служит шлюзом между ней и вашей локальной сетью. Если на любом из компьютеров вы получите с **tun** ошибку «устройство неизвестно», загрузите модуль **tun** командой

```
sudo modprobe tun
```

Хотя это и сработает, поскольку создаст подключение к одному из компьютеров сети, удобства тут мало, так как нужно, чтобы кто-то выполнил эту команду на каждом конце соединения, и должны быть заранее известны IP-адреса, а это не сильно вам поможет при подключении с помощью 3G-модема или открытого Wi-Fi во всяких местах общепита.

Наш ответ

Решение – постоянная (или, по крайней мере, в то время, когда он вероятнее всего будет необходим) работа сервера в вашей локальной сети, чтобы обеспечить соединение с любого IP-адреса и использовать сертификаты для аутентификации всех компьютеров, которые пытаются подключиться. Большинство настроек выполняется на сервере, в терминале, от имени суперпользователя-root (пользователям Ubuntu надо будет все команды предвелять **sudo**). Ваша установка *OpenVPN* должна содержать скрипты для этого в */usr/share/openvpn/easy-rsa*. Нам надо их отредактировать, так что скопируйте каталог *easy-rsa* в */etc/openvpn*, чтобы пресечь порчу ваших настроек каким-нибудь обновлением. Отредактируйте файл *vars*, задав в нем переменные **KEY_*** – ни одной не оставляйте пустой. Следующий шаг – создание мастер-сертификата удостоверяющего центра (Certificate Authority, CA) при посредстве

```
source ./vars/
./clean-all
./build-ca
```

Вам зададут несколько вопросов, но будет нормально просто нажать Enter в ответ на каждый, поскольку ответы уже указаны

► Создавайте сертификаты для сервера и клиентов, используя скрипты сборки, предоставленные *OpenVPN*.

```
root@hactar:~/openvpn/easy-rsa# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/keys
root@hactar:~/openvpn/easy-rsa# ./build-key-pass krikkit
Generating a 1024 bit RSA private key
.....++++++
writing new private key to 'krikkit.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
.....
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [UK]:
State or Province Name (full name) [Cheshire]:
Locality Name (eg, city) [Warrington]:
Organization Name (eg, company) [Digital Media Production]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [krikkit]:
Name []:me
Email Address [neil@digimed.co.uk]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:

```


в файле **vars**. В каталоге ключей **keys** создается сертификат, используемый для подписи всех серверных и клиентских сертификатов, которые мы собираемся создать. Изготовим сертификат сервера командой

```
./build-key-server имя_сервера
```

Как и прежде, примите значения по умолчанию, но задайте Common Name [общее имя] как имя сервера, а затем ответьте у на вопрос, хотите ли вы подписать и применить сертификат. Теперь запустите эту команду для каждого клиента, которому вы хотите предоставить подключение, используя одно и то же имя клиента в командной строке и в ответе на вопрос про Common Name:

```
./build-key имя_клиента
```

Этот скрипт создает ключ, разрешающий подключение к VPN, так что любой, кто имеет доступ к компьютеру, сможет его получить. Если это ноутбук, и у вас нет других форм защиты от вора, используйте **build-key-pass** вместо **build-key** для создания ключа, прикрытого паролем.

Надо будет построить еще один файл, командой

```
./build-dh
```

Скопируйте файл **ca.crt** и файлы **.crt** and **.key** этого компьютера в каталог **/etc/openvpn** каждого компьютера. На сервере также необходимо скопировать файлы **ca.key** и **dh1024.pem**, созданные командой **build-dh**. Сделайте это безопасно, с помощью сменного носителя либо SSH – не посылайте их по электронной почте, поскольку с этими файлами к вашей сети сможет подключиться любой. Задайте разрешения на секретный файл **.key file** как **rw-----**. Не забудьте, что копирование их на USB-брелок с форматом FAT установит их в гораздо более либеральные, чем неприемлемо.

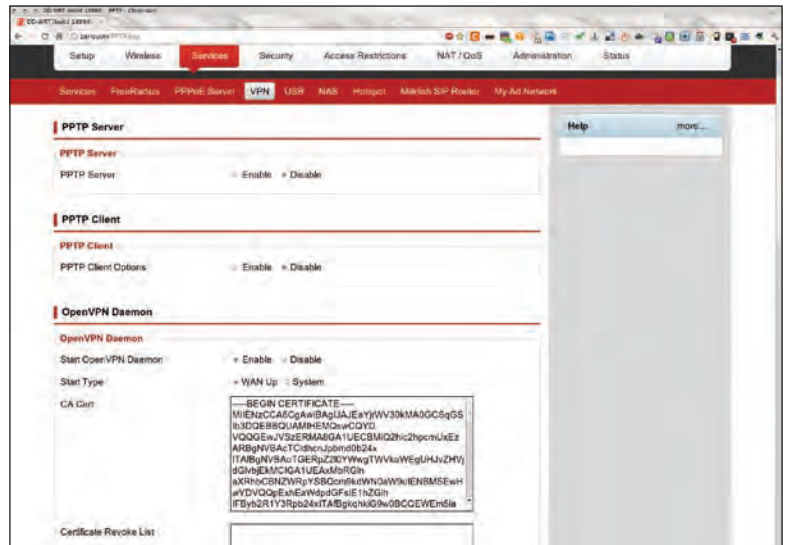
И серверу, и клиенту нужен файл конфигурации в **/etc/openvpn**. Файлы-примеры, обычно помещаемые в **/usr/share/doc/openvpn/examples/sample-config-files**, будут лучшей отправной точкой. Скопируйте файл **client.conf** в **/etc/openvpn/ openvpn.conf** и отредактируйте его на сервере. Большинство строк файла являются комментариями, и настройки по умолчанию хороши, но убедитесь, что параметры **ca**, **cert**, **key** и **dh** соответствуют созданным вами файлам, желательно с указанием полного пути.

Скопируйте файл **client.conf** в **/etc/openvpn/openvpn.conf** на клиент и отредактируйте его. Измените строку **remote**, чтобы она указывала на IP-адрес (или имя) вашего сервера и открытый на нем порт (по умолчанию, 1194). Вы можете иметь и не одну такую строку – в этом случае они будут опрашиваться по очереди, пока не установится соединение.

```
remote gateway.example.com 1194
```

```
remote 123.124.125.126 1194
```

Используйте публичный IP-адрес вашего интернет-соединения. Если ваш шлюз находится за маршрутизатором, пробросьте порт 1194 на сервер OpenVPN в настройках маршрутизатора. Измените



строки **cert** и **key**, чтобы они содержали имена сертификата клиента и файлы ключей. Если вы меняли параметры по умолчанию в настройках сервера, убедитесь, что все настройки здесь им соответствуют. Теперь вы увидите, как все это работает, запустив

```
/etc/init.d/openvpn start
```

на сервере, а затем на клиенте. Запуск **ifconfig** на каждом компьютере должен показать интерфейс **tun** с адресом в диапазоне **10.8.0.*** (если вы не меняли этого в настройках), и вы должны быть в состоянии pingовать между ними. Убедившись, что это работает, вы можете воспользоваться услугами менеджера сервисов вашего дистрибутива, чтобы настроить автоматический запуск **OpenVPN** при загрузке.

Добраться до всей сети

Пока что мы соединили два компьютера, но мы хотим, чтобы клиент мог получить доступ к всей локальной сети, а не только к серверу. **OpenVPN** изменил таблицу маршрутизации на клиенте, пропустив весь трафик сети **10.8.0.0** через VPN. Чтобы сервер направлял трафик для локальной сети правильным получателям, добавьте в его файл настройки

```
push "route 192.168.1.0 255.255.255.0"
```

используя соответствующий адрес и маску для локальной сети. Тогда не только установится маршрутизация на сервере, но сервер использует это для отправки правильных настроек маршрутизации на клиент при его подключении. Если вы скомандуете **route -n** после открытия VPN на клиенте, он покажет нечто вроде

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	10.8.0.2	255.255.255.0	UG	0	0	0	tun0

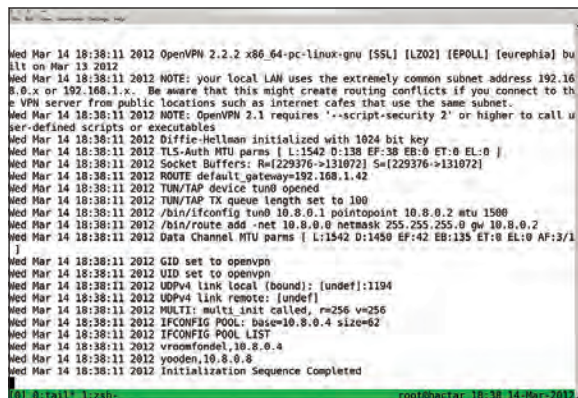
Это будет хорошо работать, если ваш сервер **OpenVPN** служит также и шлюзом для сети, то есть другие компьютеры в локальной сети будут посылать на него весь не-LAN трафик в любом случае. В противном случае у вас есть два варианта. Вы можете изменить таблицы маршрутизации у всех компьютеров в сети, к которым вы хотите иметь доступ через VPN, выполнив на них команду

```
route add -net 10.8.0.0 netmask 255.255.255.0 gw 192.168.1.1
```

или можно изменить настройки маршрутизатора, направив трафик в сети **10.8.0.0** через сервер – в данном примере, **192.168.1.1**. Некоторые маршрутизаторы, особенно те, что работают на ОС DD-WRT или OpenWrt, поддерживают **OpenVPN** напрямую. Если у вас есть такой маршрутизатор, переложите задачу на него, с помощью web-интерфейса.

Детали различаются в зависимости от прошивки маршрутизатора, но в основном это вставка содержимого вашего сертификата, ключа и файлов конфигурации в текстовые поля web-интерфейса маршрутизатора. **LXF**

➤ На маршрутизаторе с поддержкой OpenVPN, подобном устройствам на базе DD-WRT, можно запустить сервер, избежав необходимости всегда иметь включенным один конкретный компьютер.



➤ При тестировании запускайте сервер и клиент с терминала. Это позволит вам точно знать, что происходит.



LaTeX. Массивы

Дмитрий Цымай и Станислав Куценко предлагают разрабатывать тестовые задания для студентов средствами LaTeX, применив их для подготовки текстов.

Наш эксперт

Станислав Куценко

д. т. н., профессор ФГБОУ ВПО «Госуниверситет-УНПК». Оказывает, заведование кафедрой химии не ограничивает кругозор этой наукой.

Наш эксперт

Дмитрий Цымай

к. т. н., доцент ФГБОУ ВПО «Госуниверситет-УНПК». Кинетика гетерогенных систем привела его к системам типографским.

LaTeX – популярный набор макрорасширений системы компьютерной верстки TeX, облегчающий подготовку сложных документов. Применение макропакетов позволяет автоматизировать задачи набора текста и подготовки статей. LaTeX расширяем также за счет написания собственных макрокоманд – например, при помощи встроенной в TeX команды `\def`, описанной в [1]. Чтобы случайно не переопределить существующую команду, можно пользоваться встроенными в LaTeX командами `\newcommand` для определения новой команды и `\renewcommand` для явного переопределения существующей, описанными в [2].

Все это удобно до тех пор, пока не потребуется определить достаточно много новых макрокоманд. Такая задача встала при разработке тестовых заданий для студентов. Одни и те же вопросы могут повторяться в разных тестовых заданиях, и разумно определить для каждого из них макрокоманду: это сократит время и на верстку заданий, и при правке вопросов. Но сложность в том, что каждая макрокоманда требует уникального имени, а цифры в именах использовать нельзя; необходимость держать в памяти множество новых команд сводит к нулю все преимущества их повторного использования. При верстке тестов необходимо иметь способ вставить фрагмент (отдельный вопрос) в произвольное задание, не вспоминая при этом имя командной последовательности. В [1] описан вариант реализации списочных макрокоманд. Данный подход не совсем удобен: он предполагает сохранение всех фрагментов в определении одной списочной макрокоманды.

Одно из возможных решений – массив макрокоманд. Достаточно определить одно имя массива для фрагментов текста, разметить каждый вопрос (начало и конец) определенными ограничивающими командами и в дальнейшем обращаться к данным фрагментам по номерам, в соответствии с порядком их определения в тексте. Присвоение имен новым командам перекладывается на компьютер, т. е. происходит прозрачно для пользователя.

Основные команды для массивов

LaTeX не имеет своих средств для работы с массивами макрокоманд. Встроенные средства программирования [3] предлагают:

- » объявление массива с произвольным именем;
- » добавление в массив произвольных элементов;
- » перебор всех или части элементов массива в цикле с выполнением над элементами операций форматирования.

Разберем особенности реализации. Раз имя макрокоманды не должно включать арабских цифр, заменим их на римские, отображаемые латинскими буквами. Тогда серия команд `\query1`, `\query2`, `\query3...` запишется `\queryi`, `\queryii`, `\queryiii...`. Каждое имя будет уникальным, но строка перед римскими цифрами не должна совпадать с существующими командами LaTeX.

Команда `\newArray<имя массива>` определяет новый массив, и автоматически создает счетчик массива с именем `arraycount<имя определяемого массива>` и начальным значением 1. Параметр команды – имя нового массива, подчиняющееся тем же правилам, что и имена счетчиков и макрокоманд. Определение команды –

```
\def\newArray#1{
  \newcounter{arraycount#1}
  \setcounter{arraycount#1}{1}
}
```

Для объявления нового массива необходимо записать:

```
\newArray<имя массива>{
  \AddItem<имя массива><элемент>
}
```

Команда `\AddItem<имя массива><элемент>` позволяет добавлять элементы в объявленный массив. Каждый элемент будет храниться в новой команде, с именем `array<имя массива><значение счетчика римскими цифрами>`. Так обеспечивается уникальность имени каждого элемента. Например, для массива с именем `query` имена макрокоманд для 1-го, 2-го и 3-го элементов будут, соответственно, `arrayqueryi`, `arrayqueryii` и `arrayqueryiii`. При выполнении команды счетчик, связанный с массивом, увеличивает на единицу. Команда имеет следующее определение:

```
\long\def\AddItem#1#2{%
  \expandafter\gdef\csname array#1\roman{arraycount#1}\endcsname{#2}
  \global\addtocounter{arraycount#1}{1}
}
```

Рассмотрим элементы TeX в составе определения. Команда `\csname... \endcsname`

задает новую командную последовательность без содержания; позже ее можно переопределить. Команда `\expandafter` велит TeX'у раскрыть последовательность, определяющую имя новой макрокоманды, а затем определить ее глобально при помощи `\gdef`. Команда `\roman{arraycount#1}` выводит значение счетчика, связанного с массивом римскими цифрами. Общий синтаксис команды – `\roman<имя счетчика>`.

Элементы массива по его номеру извлекается командой

```
\GetItem<имя массива>{<элемент>}
имеющая два параметра: имя массива и номер элемента.
\long\def\GetItem#1#2{%
  \setcounter{temparraycount}{#2}
  \ifnum\arabic{temparraycount}<\arabic{arraycount#1}
    \csname array#1\roman{temparraycount}\endcsname
  \fi
}
```

Здесь используется вспомогательный счетчик `temparraycount`, определенный ранее. Условный оператор `\ifnum<Команды>`, позволяет выполнять команду, если номер вызываемого элемента не выходит за границы массива. Выполняется сравнение значений счетчиков `temparraycount` и `arraycount<имя массива>`, представленные арабскими цифрами, командой `\arabic<имя счетчика>`.

Число элементов массива на 1 меньше значения счетчика `arraycount#1`. Определение для команды `\ArrayCount<имя массива>` получения числа элементов массива –

```
\def\ArrayCount#1{
  \setcounter{temparraycount}{\arabic{arraycount#1}}
  \global\addtocounter{temparraycount}{-1}
  \arabic{temparraycount}
  \global\addtocounter{temparraycount}{1}
}
```

Также необходимы команды перечисления всех элементов массива. Их две. Одна является аналогом команды `foreach... in...`. Она принимает два параметра: имя массива и команду, которую надо исполнить над каждым элементом при его выводе.

МАКРОКОМАНДА

```
\long\def\foreachArray#1#2{%
\setcounter{temparraycount}{1}
\loop\ifnum\arabic{temparraycount}<\arabic{arraycount#1}
#2\GetItem{#1}\arabic{temparraycount}}
\global\addtocounter{temparraycount}{1}
\repeat
}
```

Подобна ей команда-аналог цикла **for... from... to**. У нее 4 параметра: имя массива, номера первого и последнего выводимого элемента и операция, исполняемая над элементами при выводе.

```
\long\def\ForFromToArray#1#2#3#4{%
\setcounter{temparraycount}{#2}
\setcounter{temparraycounta}{#3}
\ifnum\arabic{temparraycounta}<\arabic{arraycount#1}
\addtocounter{temparraycounta}{1}
\else
\setcounter{temparraycounta}{\arabic{arraycount#1}}
\fi
\loop\ifnum\arabic{temparraycount}<\arabic{temparraycounta}
#4\GetItem{#1}\arabic{temparraycount}}
\global\addtocounter{temparraycount}{1}
\repeat
}
```

В двух последних определениях команда `\loop\if <условие><тело цикла>... \repeat` позволяет использовать цикл. Тело цикла выполняется, пока истинно условие после `\if`.

Массивы команд в действии

Рассмотрим примеры применения предложенных команд. Для этого в преамбуле документа подключим файл с определениями:

```
\input{texarray}
```

Определяем два массива: **mytestq** и **mytestw**.

```
\newArray{mytestq} \newArray{mytestw}
```

Добавляем элементы в оба массива.

```
\AddItem{mytestq}{фрагмент 1}
\AddItem{mytestw}{фрагмент 2}
\AddItem{mytestq}{фрагмент 3} \
AddItem{mytestw}{фрагмент 4}
\AddItem{mytestw}{фрагмент 5}
\AddItem{mytestw}{фрагмент 6}
\AddItem{mytestw}{фрагмент 7}
```

Выводим второй элемент массива **mytestq**: фрагмент 3.

```
\GetItem{mytestq}{2}
```

Количество элементов в массиве **mytestq** равно 2.

```
\ArrayCount{mytestq}
```

Количество элементов в массиве **mytestw** равно 5.

```
\ArrayCount{mytestw}
```

Перечисляем все элементы массива **mytestw**:

```
\foreachArray{mytestw}{}
```

Результат: фрагмент 2 фрагмент 4 фрагмент 5 фрагмент 6 фрагмент 7

Добавляем еще один элемент в массив **mytestq**:

```
\AddItem{mytestq}{фрагмент 8}
```

Перечисляем все элементы массива **mytestq**

```
\foreachArray{mytestq}{}
```

Результат: фрагмент 1 фрагмент 3 фрагмент 8

Перечислим все элементы массива **mytestq** с выполнением операции, заданной вторым параметром, над всеми элементами массива. Для этого определим команду вывода текста жирным курсивом –

```
\def\testcommand#1{\textbf{\textit{#1}}}
```

и применим ее в вызове функции перебора элементов массива.

```
\foreachArray{mytestq}{\testcommand}
```

Результат: **фрагмент 1 фрагмент 3 фрагмент 8**

Перечислим элементы массива **mytestw** со 2-го по 4-й.

```
\ForFromToArray{mytestw}{2}{4}{}
```

Результат: фрагмент 4 фрагмент 5 фрагмент 6

Выведем элементы массива **mytestw** со 2-го по 4-й с выполнением операции над каждым элементом массива.

```
\ForFromToArray{mytestw}{2}{4}{\testcommand}
```

Результат: **фрагмент 4 фрагмент 5 фрагмент 6**

Пробуем создать тесты

Приведенные ранее команды являются достаточно общими, что позволяет использовать их в составе других команд. Рассмотрим одно из их практических применений для верстки тестовых заданий.

Счетчики и вспомогательные определения

Для работы с макрокомандами определим несколько счетчиков:

- **BiletNumber** Счетчик номер билета
- **VoprosNumber** Счетчик для вопросов
- **VarOtvNumber** Счетчик вариантов ответов

Установим указанные счетчики равными 1 при помощи команд

```
\newcounter{BiletNumber}
\setcounter{BiletNumber}{1}
\newcounter{VoprosNumber}
\setcounter{VoprosNumber}{1}
\newcounter{VarOtvNumber}
\setcounter{VarOtvNumber}{1}
```

Команды для разметки исходных вопросов

Для работы с тестами мы используем созданные нами команды работы с массивами. Определим новый массив:

```
\newArray{qvery}
```

Определим команду для обозначения вопроса, добавляющую текстовый фрагмент в массив **qvery**, основанную на команде **\AddItem**:

```
\long\def\Qvery #1\endQvery{%
\AddItem{qvery}{#1}
}
```

К вопросам, обозначенным при помощи `\Qvery #1\endQvery`, можно обратиться по номеру, соответствующему их появлению в тексте, при помощи другой команды – `\printQ<номер вопроса>`, основанной на использовании команды `\GetItem{qvery}{<номер вопроса>`.

```
\long\def\printQ#1{%
\setcounter{VarOtvNumber}{1}
\par\noindent\arabic{VoprosNumber}. % \
GetItem{qvery}{#1}
\addtocounter{VoprosNumber}{1}%
}
```

В результате печатается элемент массива **qvery** с номером, координатным параметром. Пример выполнения команды показан на рис. 1.



1. 6-й вопрос

Рис. 1. Отдельный вопрос, выведенный командой printQ.

```
\long\def\VarQveryOtv{%
\par\indent\ \arabic{VarOtvNumber}) %
\addtocounter{VarOtvNumber}{1}%
}
```

Команды, определяющие вид печатного документа

Эти команды имеют вспомогательное значение. Они могут быть использованы при необходимости.

Значение переменной **printotv** определяет, помечать ли в конечном документе правильные ответы, если используются команды для обозначения правильных вариантов ответов. Если оно равно 1, правильные ответы отмечаются. Значение **prnsimple** определяет, печатать ли все заголовки, определенные для теста. Если оно равно 1, печатаются только основные заголовки.

```
\newcount\printotv
\printotv=0 \
newcount\prnsimple
\prnsimple=0
```

Значения этих переменных для любой части документа задаются командами

- ▶ **\printsimple** Печатать без заголовков.
- ▶ **\printall** Печатать с заголовками.
- ▶ **\setprintrueotv** Помечать правильные ответы на печати
- ▶ **\setnoprintrueotv** Не помечать правильные ответы на печати
- ▶ **\trueotv** Маркер правильных ответов.

Указанные команды действительны для оставшейся части текста, пока не встретится отменяющая команда. Приведем определения этих команд:

```
\def\printsimple{\prnsimple=1}
\def\printall{\prnsimple=0}
\def\setprintrueotv{\printotv=1}
\def\setnoprintrueotv{\printotv=0}
\def\trueotv{\ifnum\printotv=1$\bullet $ \fi}
```

Кроме того, вид печатного документа определяют переменные шаблона, определяющие последующий текст, которые можно задать в любом месте документа. Дальнейшее рассмотрение ведется на примере экзаменационных билетов. Приведенные ниже команды позволяют определить ряд строк в заголовке – например, в экзаменационных билетах это строки «Утверждаю», «Предмет», «Специальность», «Зав каф», «Составил». Данные команды определяют, в свою очередь, новые команды, используемые внутри шаблона.

```
\def\templateutv#1{\edef\ltemplateutv{#1}}
\def\templatepredm#1{\edef\ltemplatepredm{#1}}
\def\templatespez#1{\edef\ltemplatespez{#1}}
\def\templatezavkaf#1{\edef\ltemplatezavkaf{#1}}
\def\templatesost#1{\edef\ltemplatesost{#1}}
```

Аналогично определяется ряд других параметров теста, таких как название подразделения, фамилия руководителя, дата утверждения, название организации, фамилия составителя, заголовок теста, название предмета, шифр специальности.

```
\def\kafedra#1{\edef\lkafedra{#1}}
\def\zavkaf#1{\edef\lzavkaf{#1}}
```

```
\def\datautv#1{\edef\ldatautv{#1}}
\def\vuz#1{\edef\lvuz{#1}}
\def\biletttitle#1{\edef\lbiletttitle{#1}}
\def\biletauthor#1{\edef\lbiletauthor{#1}}
\def\predmet#1{\edef\lpredmet{#1}}
\def\spez#1{\edef\lspez{#1}}
```

Переопределение строк шаблона можно сделать в любом месте документа; оно имеет силу, пока не встретится новое определение.

Выполняется начальная инициализация переменных до определения шаблонов и окружений, где они используются.

```
\templateutv{УТВЕРЖДАЮ}
\templatepredm{Предмет}
\templatespez{Специальность}
\templatezavkaf{Зав.кафедрой}
\templatesost{Составил}
\kafedra{<Название кафедры>}
\zavkaf{<Фамилия И.О.>}
\datautv{<чч.мм.гг>}
\vuz{<Название ВУЗа>}
\biletttitle{<Заголовок билета>}
\biletauthor{<Фамилия И.О.>}
\predmet{<Название предмета>}
\spez{<Специальности>}
```

Команды для компоновки отдельных тестов

Для работы с отдельными тестами предусмотрена команда для заголовка теста **\BiletHeader**. Результат ее выполнения зависит от переменных шаблона. Происходит установка счетчика номеров вопросов **VoprosNumber** равным 1, а счетчик номеров билета **BiletNumber** увеличивается на 1.

```
\newcommand{\BiletHeader}{%
\ifnum\prnsimple=0
\nopagebreak \par
\noindent\textbf{\ltemplateutv}
\space{0.4cm}\nopagebreak
\par
\ltemplatezavkaf <<\lkafedra >>
\par%
\rule[-1pt]{2 cm}{0.4pt} \lzavkaf
\space{0.2cm}\nopagebreak
\par%
\ltemplatepredm : <<\lpredmet >>\nopagebreak \par
\ltemplatespez : \lspez\nopagebreak \par
\datautv
\space{0.1cm}\nopagebreak \par
\begin{center}
\textbf{\lvuz}\nopagebreak
\end{center}
\fi
\begin{center}
\textbf{\lbiletttitle \No \arabic{BiletNumber}}\nopagebreak
\space{0.2cm}\nopagebreak
\addtocounter{BiletNumber}{1}
\setcounter{VoprosNumber}{1}
\nopagebreak
\end{center}
}
```

Команда для нижней части теста **\BiletFooter**. Здесь только команды форматирования, выводимые в зависимости от условий.

```
\newcommand{\BiletFooter}{%
\ifnum\prnsimple=0
\space{0.4cm}\nopagebreak \par
\noindent\ltemplatesost
\rule[-1pt]{1.5cm}{0.4pt}
\biletauthor \nopagebreak
```



Рис. 2. Список вопросов, выведенный командой makelistquery.

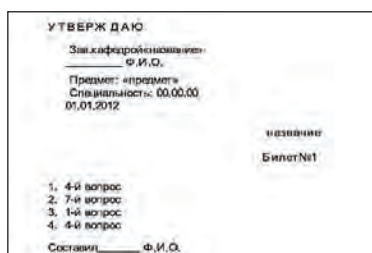


Рис. 3. Билет, распечатанный при помощи окружения EkzBilet или командой printbilet.

```
\vspace{0.1cm}
\fi
\par
}
```

Окружение для билета, в котором используются обе вышеприведенные команды для заголовка и нижней части билета – `\newenvironment{EkzBilet}{\BiletHeader}{\BiletFooter}`

Окружение для списка вопросов –

```
\newenvironment{ListQvery}{%
\par\noindent\textbf{\templateutv}
\vspace{0.4cm}
\\
\ \templatezvaf <<\kafedra >> \rule[-1pt]{3cm}{0.4pt} \zvafkaf
\vspace{0.2cm}
\\
\begin{center}
```

```
\textbf{Список вопросов по курсу <<\lpredmet >>}\
для специальности \spez
\end{center}
\setcounter{VoprosNumber}{1}
}{\BiletFooter}
```

Данное окружение использовано в команде для распечатки списка вопросов.

```
\def\makelistqvery{%
\begin{ListQvery}
\setcounter{VoprosNumber}{1}
\foreachArray{qvery}{\par\noindent\arabic{VoprosNumber}.%
\addtocounter{VoprosNumber}{1}}
\end{ListQvery}
}
```

Используется команда, распечатывающая все элементы массива `qvery`. Перед распечаткой очередного элемента выполняется команда распечатать порядковый номер вопроса и увеличить его на 1.

```
\par\noindent\arabic{VoprosNumber}. \
\addtocounter{VoprosNumber}{1}
```

Результат работы команды представлен на рис. 2. Для компоновки отдельного теста можно воспользоваться как вышеприведенными окружениями, так и командой `\printbilet`, параметром которой являются номера вопросов, включаемых в тест, разделенные командой `\and`. Результат работы команды представлен на рис. 3.

```
\long\def\printbilet #1{
\let\and=\printQ %
\begin{EkzBilet}
\printQ #1
\end{EkzBilet}
}
```

Команды для компоновки тестов

Для компоновки блоков из 2-х и 4-х тестов используется окружение `tabular`. Определена команда `\twotestblok` для компоновки блока из 2-х тестов, которая имеет два параметра, задаваемых по аналогии с параметром команды `\printbilet`. Результат работы команды показан на рис. 4.

```
\long\def\twotestblok#1#2{
\begin{tabular}{p{0.9\linewidth}}
\hline \ \printbilet{#1}\ \hline \ \printbilet{#2} \ \hline %
\end{tabular}
}
```

Определена команда `\fiertestblok` для компоновки блока из 4-х тестов, которая имеет четыре параметра, задаваемых по аналогии с параметром команды `\printbilet`. Результат ее работы – на рис. 5.

```
\long\def\fiertestblok#1#2#3#4{
\small
```



Рис. 4. Блок из двух билетов, полученный при помощи команды `twotestblok`.



Рис. 5. Блок из четырех билетов, полученный при помощи команды `fiertestblok`.

```
\begin{tabular}{p{0.45\linewidth}p{0.45\linewidth}}
\hline \ \printbilet{#1} & \printbilet{#2} \ \hline \ \ %
\printbilet{#3} & \printbilet{#4} \ \hline
\end{tabular}
}
```

Все рассмотренные выше команды можно свести в отдельный файл `testsmakr`, подключаемый командой `\input{testsmakr}`. Пример рабочего файла, при компиляции которого получены результаты, представленные на рисунках, мы предоставляем собрать читателю. (Те, кто хочет убедиться в правильности своего результата, найдут полный текст на стр. 111.)

В данном примере определены десять вопросов, которые используются при выводе тестов при помощи рассмотренных в статье команд.

Заключение

Вместо множества команд для вставки каждого фрагмента пользователю достаточно запомнить лишь пять:

- 1 Команды– ограничители начала и конца запоминаемого фрагмента
- 2 Команды вставки отдельного фрагмента
- 3 Команды вставки всех фрагментов или их части
- 4 Команды вставки блоков из двух или четырех билетов
- 5 Два новых окружения, специализированных для подготовки тестов: список вопросов и билет

Команды и окружения, указанные в 4-м и 5-м пунктах, в отличие от пунктов 1–3, специализированы для подготовки тестовых заданий. Кроме того, при выводе всех фрагментов имеется возможность выполнения произвольного форматирования над каждым фрагментом.

Преимуществом подобного подхода является возможность для автора сосредоточиться на содержании самих вопросов, а не на их оформлении при компоновке тестов. Значительно упрощается корректировка вопросов. Править необходимо только в одном месте без использования поиска и автозамены, как при использовании копирования – вставки из буфера обмена в MicrosoftWord.

Результатом использования рассмотренных в статье макрокоманд является повышение эффективности работы за счет реложения на компьютер рутинных задач. **LXF**

Список литературы

- 1 Кнут Д. Все про TeX. – Протвино, RDTEX, 1993.
- 2 Львовский С. М. Набор и верстка в пакете LaTeX. – М., Космосинформ, 1994.
- 3 <http://www.tex.uniyar.ac.ru/> – TeXв ЯрГТУ

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru, и мы найдем ответ.

» В этом месяце мы ответим на вопросы про...

- 1 Выключение системы в VirtualBox
- 2 Sent OS и мобильный широкополосный доступ
- 3 Замедление процесса загрузки
- 4 Передачу данных при посредстве rsync
- 5 Ошибку, закрывающуюся в DVD LXF
- 6 Увеличение размера раздела

1 Виртуальное выключение

В Я виртуализировал стек LAMP на базе XP Pro. Я запускаю эту систему в качестве гостевой на SUSE 12.1 с *VirtualBox 4.1.8* и KDE 4.7.2. Мне нужно гарантировать, чтобы вся система, включая виртуальный сервер, запускалась при включении питания и корректно выключалась. При включении питания я запускаю ее в KDE с помощью `auto-login`. Для автоматического запуска гостевой ОС на моем рабочем столе я поместил скрипт в `~home/myname/.kde4/Autostart`. Пока все хорошо, и он работает. Я не запускаю его «без головы», так как хочу иметь быстрый доступ к рабочему столу сервера XP Pro. Теперь о проблеме. При попытке включить систему или хотя бы выйти из системы, *VirtualBox* спрашивает, что я хочу

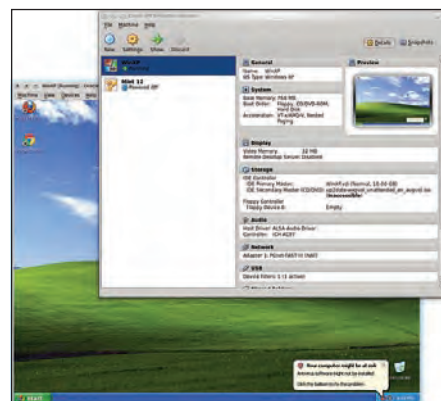
сделать с гостевой системой, и тут же прекращает работу. А мне нужно корректно завершить работу гостевой ОС, так как в ней большая база данных *MySQL*.

Я пытался добавить скрипт в `~home/myname/.kde4/shutdown`, но похоже, что *VirtualBox* опережает его в цепочке событий выключения. Как заставить KDE завершить работу системы или выйти из системы так, чтобы *VirtualBox* понял, что нужно отправить стандартный сигнал выключения ACPI гостевой ОС? На первый взгляд это просто, но я уверен, что это не так.

Дэвид Гоудби [David Goadby]

Вам нужна *VBoxManage*, утилита командной строки, с помощью которой можно управлять *VirtualBox* с терминала или, в данном случае, запускать скрипт выключения системы.

Напрямую отправить команду выключения виртуальной машине нельзя, хотя с помощью опции `guestcontrol` можно запустить в гостевой системе команду, которая сделает то, что вам нужно. Однако в виртуальных системах удобно то, что их не обязательно выключать – можно просто сохранить их состояние. Это аналогично переводу ноутбука в спящий режим – состояние



» GUI *VirtualBox* упрощает жизнь в интерактивном режиме, или можно управлять VM из скриптов.

гостевой системы сохраняется и восстанавливается при следующем запуске. Этого должно быть достаточно, к тому же это ускорит процесс.

Для начала определите свои виртуальные машины, командой

```
VBoxManage list vms
```

которая выведет примерно такой список:

```
"WinXP" {d546c405-6a1c-4bb9-9ffb1ccec59dfdd}
"Mint 12" {4d7daa80-5473-4454-b974-7048f5c4394c}
```

Виртуальную машину можно определить по ее имени или по UUID (длинная строка в фигурных скобках), но если используется имя, и в нем есть пробелы, оно должно быть заключено в кавычки.

Теперь можно приостановить виртуальную машину XP командой

```
VBoxManage controlvm "WinXP" savestate
```

или

```
VBoxManage controlvm d546c405-6a1c-4bb9-9ffb1ccec59dfdd savestate
```

Действительно умно было бы написать скрипт, приостанавливающий все запущенные виртуальные машины.

```
for VM in $(VBoxManage list runningvms |
```

```
awk '{print $NF}'); do
```

```
VBoxManage controlvm $VM savestate
```

```
done
```

Приостановленную виртуальную машину можно перезапустить обычным образом или с помощью *VBoxManage* в скрипте запуска:

```
VBoxManage startvm "WinXP"
```

Для этого не нужно запускать графический интерфейс *VirtualBox*: *VBoxManage* запустит виртуальную машину напрямую. Вы уже обнаружили, что при выходе из системы KDE запускает все из `~/.kde4/shutdown`. Решение – воспользоваться скриптом выключения системы, который со-

Наши эксперты

» Мы подыщем эксперта на любой ваш вопрос. От установки и проблем с модемом до администрирования сети – ответ найдется! Просто напишите нам бумажное или электронное письмо, и о нем позаботятся.



Нейл Ботвик

Побыв владельцем ISP и редактором дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс

Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Джонатан Робертс

Джонатан столько отлынивал от занятий ради установки, настройки и взлома серверов, что теперь он эксперт.



Грэм Моррисон

Когда Грэм не обозревает кучи программного обеспечения и не халтурит с *MythTV*, он готов к ответам насчет любого оборудования и проблем виртуализации.



Валентин Синецын

В редкие свободные минуты Валентин обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.



Юлия Дронова

Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования www.unixforum.org.

Куда посылать вопросы

Пишите нам по адресу answers@linuxformat.ru или спрашивайте на форуме: www.linuxformat.ru

хранит виртуальную машину и затем велит KDE выйти из системы.

```
VBoxManage controlvm "WinXP" savestate
qdbus org.kde.ksmserver /KSMserver logout 0 0 0
```

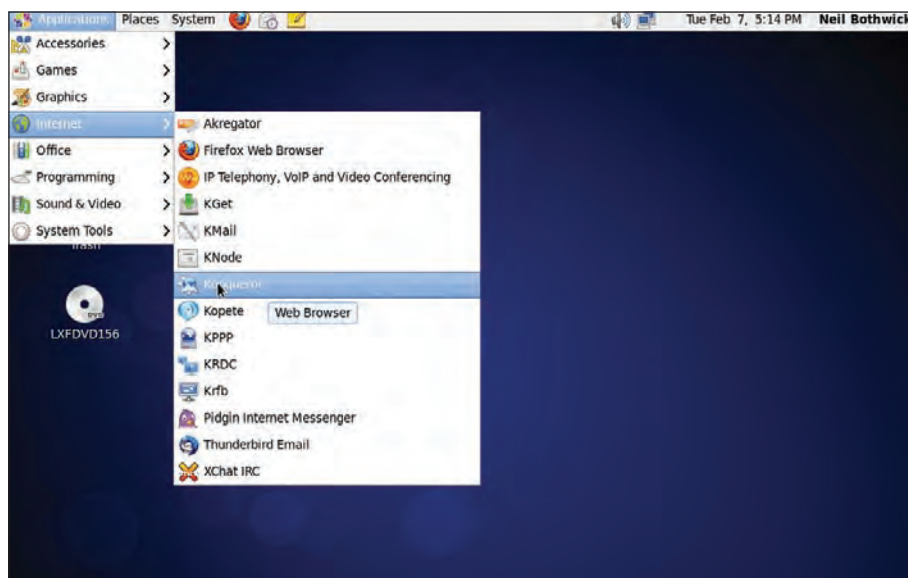
Может понадобится вставить команду sleep между этими двумя командами, чтобы дать виртуальной машине время на сохранение. Если открыт графический интерфейс VirtualBox, его также можно закрыть (если больше ничего не поможет, killall VirtualBox сделает эту работу). Команда qdbus отправляет KDE запрос на выход из системы. Если вы хотите завершить работу системы, а не выйти из нее, измените 0 0 0 на 0 2 2. Можете связать этот скрипт с горячей клавишей, с пунктом меню или с иконкой на рабочем столе, дав им подходящее имя.

2 Мобильный 3G-доступ

Недавно я установил CentOS 6 с одного из ваших DVD на свой ноутбук с двойной загрузкой с Windows Vista. У меня есть 3G-модем от T-Mobile, который в Windows Vista настроился автоматически. Как заставить его работать в CentOS?

Недавно в Linux Format появилось письмо про DVD, прилагаемый к журналу. Пожалуйста, пожалуйста, не отказывайтесь от DVD. Возможности загрузки файлов по 3G-соединению просто жалкие, и целый дистрибутив загрузить трудно.

Кен Урвин [Ken Urwin]



CentOS 6 на нашем невероятно успешном LXF DVD с номером 156.

Это распространенная проблема, и с CentOS с ней придется повозиться немного больше. USB-модемы используют специальный прием, чтобы не пользоваться драйвером CD в Windows. Когда вы вставляете флешку, она определяется как компакт-диск, с которого запускается установщик для Windows.

После установки драйверы определяют один из этих псевдо-CD и превращают его в модем. Это можно увидеть в Linux, когда при подключении модема обнаруживается новый компакт-диск, которому соответствует не последовательное устройство модема, а /dev/ttyUSB0. Если этого устройства не существует, нужно приказывать модему, чтобы он перестал вести себя как CD и стал самим собой. Обычно это можно сделать вручную, попытавшись извлечь компакт-диск. Этому компакт-дису обычно соответствует устройство /dev/sr1 (/dev/sr0 – настоящий привод CD), поэтому с помощью команды

```
sudo eject /dev/sr1
```

вы сможете избавиться от CD и увидеть устройство /dev/ttyUSB0. Вы вряд ли захотите делать это каждый раз при подключении модема, и есть программа, которая будет делать это автоматически, выполняя почти те же функции, что и драйверы Windows (драйверы являются частью ядра, поэтому их устанавливать не придется).

Эта программа – usb_modeswitch (http://www.draisberghof.de/usb_modeswitch/); в CentOS пакета для нее нет, и придется установить ее вручную. Во-первых, зайдите в Установку и удаление программ [Add/Remove Software] и убедитесь, что установлены все зависимости – вам понадобятся libusb и libusb-devel. Также понадобится установщик gcc, так как загружаемый пакет – это исходный код, требующий компиляции. Загрузите usbmodeswitch-1.2.3.tar.bz2 и usbmodeswitch-data-20120120.tar.bz2 с сайта usb_modeswitch. Это последние версии на момент написания статьи – на момент ее чтения там могут быть версии посвежее; тогда возьмите их. Распакуйте архивы и установите их, запустив в терминале следующие команды:

```
su
tar xf usb-modeswitch-1.2.3.tar.bz2
tar xf usb-modeswitch-data-20120120.tar.bz2
cd usb-modeswitch-1.2.3
make install
```



Коротко про...

Вежливость

Помните, как мама учила вас быть вежливым? То же относится и к программам на вашем компьютере. Число тактов процессора ограничено, и если одна программа захватит львиную долю их числа, другие пострадают. В Linux есть команда под названием nice. Она запускается так:

```
nice имяПрограммы
```

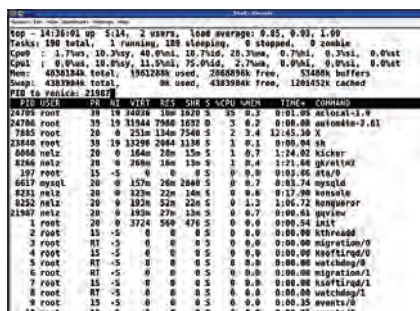
или так:

```
nice -n 5 имяПрограммы
```

Число после -n – изменение параметра nice программы. Чем выше число, тем с большим значением nice будет работать программа, и при значении nice 10, которое используется по умолчанию, если параметр команды не указан, программа будет задерживаться перед программами с меньшим значением nice, но будет иметь приоритет перед программами с большим значением nice. Значение параметра, указанное с командой nice, прибавляется к любому существующему значению nice, поэтому выполнение команды nice -n 5 в результате даст значение nice 10. Самое высокое значение nice – 19, и в этом случае циклы процессора используются программой только тогда, когда они не нужны вообще никому. Такие программы мо-

гут работать медленно, но ваша система будет откликаться с той же скоростью, как если бы они не были запущены совсем. Обычные пользователи могут только увеличивать значение nice, но root может указывать и отрицательные значения параметра. Снижая параметр, легко вывести систему из строя, так что тут необходима осторожность.

nice поможет программам с циклами процессора, но не поможет программам, требующим приоритетный доступ к жесткому диску. Для этого используйте ionice.



Программы могут использовать большое количество процессорного времени, не замедляя все остальное.

```

Started udev Kernel Device Manager.
Started udev Coldplug all Devices.
Starting udev Wait For Complete Device Initialization...
Started Setup Virtual Console.
Started udevd Wait For Complete Device Initialization.
Starting Wait for storage scan...
Started Show Plymouth Boot Screen.
Started Wait for storage scan.
Starting Initialize storage subsystems (BMD, LDM, etc.)...
Started Remount Root FS.
Starting Configure read-only root support...
Starting Initialize storage subsystems (RAID, LDM, etc.).
Starting Tell Plymouth To Write Out Runtime Data...
Starting Relabel all filesystems, if necessary.
Starting Reconfigure the system on administrator request.
Starting Enable all detected swap partitions...
Starting Setup Links in /boot for running kernel...
Starting Reconfigure the system on administrator request...
Starting Save boot message content...
Starting Cleanup of /var/run and /var/lock...
Starting Load Random Seed...
Started Tell Plymouth To Write Out Runtime Data.
Starting Load Random Seed.
Starting Enable all detected swap partitions.
Starting Initialize /etc: Started Save boot message content.
Starting Setup Links in /boot for running kernel.
The default is the current profile.

[ OK ]

Started Cleanup of /var/run and /var/lock.
Starting Recreate Volatile Files and Directories...
Starting Recreate Volatile Files and Directories failed, see 'systemctl status systemd-mpfiles-
p.service' for details.
systemd-udm (5.100.82.38.1): Already installed on this kernel.
libxobadditions (4.0.0-2): Already installed on this kernel.
Starting Reconfigure the system on administrator request.

```

➤ Выключение заставки — первый шаг в диагностике проблем с загрузкой.

```

cd ../usb-modeswitch-data-20120120
make install
exit

```

Первая команда переключает вас в режим администратора. Для установки нужны привилегии администратора; в некоторых других дистрибутивах для этого перед каждой командой придется писать `sudo`. В двух следующих строках распаковываются загруженные архивы с программой и с данными, каждый в свой собственный каталог. Затем мы заходим в каталог программы и выполняем команду `make install` — она компилирует и устанавливает программу. После этого мы переходим в каталог с данными и устанавливаем их. В последней строке мы выходим из режима администратора.

После этого все должно заработать. Вставьте USB-модем, и должно появиться устройство `/dev/ttyUSB0`, а в меню Network Manager (Менеджер сети) должна появиться возможность настройки мобильного соединения.

3 Тормозим загрузку

В последних номерах вы регулярно писали об ускорении загрузки; а не могли бы вы, для равновесия, рассказать, как замедлить загрузку? Во многих свежих системах я просто не успеваю прочесть загрузочные сообщения. В этом случае обычно советуют читать `dmesg`, но с ним есть две проблемы.

Во-первых, если загрузка останавливается в результате ошибки, `dmesg` не создается; во-вторых, при возобновлении загрузки единственное ключевое слово, которое я успеваю прочесть, когда оно мелькает на экране, кажется, никогда не попадает в `dmesg`. Иногда мне помогает параметр ядра `exit`, но нет ли более общего метода отслеживания ошибок?

Сесил Уоллис [Cecil Wallis]

О Если вследствие паники ядра загрузка полностью останавливается, то ошибка обычно содержится в нескольких последних сообщениях на экране. Панику ядра

вызывает что-то серьезное — после возникновения такой ошибки процесс загрузки далеко не уйдет. Если загрузка завершается успешно, первым делом и правда нужно взглянуть в `dmesg`. Ее вывод можно пропустить через `less`:

```
sudo dmesg | less
```

В консоли вы можете прокручивать текст назад при помощи `Shift + PageUp`, но при переключении на другую виртуальную консоль буфер прокрутки очищается — поэтому запуск X-сервера и переключение обратно с помощью `Ctrl+Alt+F1` вас не спасет.

Лучший подход при отладке такой проблемы — отключение X-сервера. Собственно говоря, лучше отключить максимально возможное количество сервисов.

Это не только позволяет выявить причину ошибки, но и очистит консоль, `dmesg` и файл системного журнала от посторонних сообщений. Если загрузка доходит до запуска системного логгера, проверьте основной лог-файл, обычно это `/var/log/messages` или `/var/log/current` в зависимости от логгера, используемого вашим дистрибутивом. Полезный прием — запустить команду:

```
ls -ltr /var/log
```

Она выведет список файлов в каталоге с лог-файлами, отсортированный по дате — нужная вам информация скорее всего будет находиться в последних файлах.

Последний вариант — войти в интерактивный режим во время загрузки. Если у вас нет заставки и вы видите загрузочные сообщения, должно появиться что-то вроде «Press I to enter interactive mode [Нажмите I для входа в интерактивный режим]».

После этого перед запуском каждого сервиса будет выдаваться запрос — это даст вам достаточно времени, чтобы прочесть вывод, и, что важнее, определить, какой сервис провоцирует проблему.

4 Массовая пересылка

В Как быстрее всего скопировать 1 ТБ данных с одного жесткого диска на другой по USB? Мне нужно скопировать видео с NAS, который решил сбросить свои настройки, но любезно отформатировал свой жесткий диск(и) в `ext3`.

Я хочу подключить жесткий диск NAS к своему MythBox через адаптер SATA →USB. Как быстрее всего скопировать эти данные?

В идеальном случае, смог бы я перевести задачу в фоновый режим, время от времени проверяя ее состояние: `rsync`, `cp` или `Ctrl+C`, `Ctrl+V` на рабочем столе в файловом менеджере *Thunar*?

bobthebob1234, с форумов

О Сразу скажем, что быстрее сделать это невозможно: скорость ограничена шиной, используемой для передачи (USB), а не программами, работающими с ней. Однако проще всего это сделать с помощью `rsync`:

```
rsync -a /path/to/source/ /path/to/dest/
```

Она копирует все из источника (`/path/to/source`) в место назначения (`/path/to/dest`), сохраняя все атрибуты, в том числе временные метки каталогов, чего нельзя добиться с помощью других способов копирования.

А главное достоинство, пожалуй, в том, что передачу с `rsync` можно возобновить, если она почему-либо прервалась. Просто запустите команду снова, и `rsync` проанализирует содержимое обоих каталогов и скопирует только файлы, не попавшие «под раздачу».

Для отображения списка копируемых файлов можно добавить параметр `-v`, это может несколько замедлить процесс, если у вас много маленьких файлов и медленный терминал (в частности, при работе по SSH). Обязательно укажите завершающие слеш, так как они определяют способ работы `rsync` с каталогами.

Если опустить их, то мы получим копию исходного каталога в каталоге назначения; указав слеш, мы копируем в каталог назначения содержимое исходного каталога — это как раз то, что нам нужно.

Ход процесса в другом терминале в любой момент можно просмотреть, запустив там команду:

```
du -sh /path/to/source /path/to/dest
```

5 Архитектурная ошибка

В При каждой моей попытке запустить *Eliot* 1.14 с `LXFDVD156` или с `web`-страницы все заканчивается сообщением «*wrong architecture [неверная архитектура]*». Это происходит в **Ubuntu 10.10** на **Lenovo X61**. Это первая ошибка, которая вообще появилась на этом компьютере. **Файл system.txt** приложен.

Билл Хекл [Bill Heckle]

О Такое сообщение обычно означает, что вы пытаетесь запустить 64-битную программу в 32-битной системе, или наоборот. Убедитесь, что вы выбрали верный пакет с DVD — `i386` для 32-битной архитектуры, `amd64` для 64-битной. Даже если установлен процессор Intel (который и есть у вас), если он 64-битный, нужно использовать пакет `amd64`.

Причины такого обозначения — исторические: AMD первой представила 64-битный процессор на основе прежней 32-битной технологии. У Intel были свои совершенно другие процессоры Itanium. Хотя сейчас Intel производит процессоры с помощью той же базовой архитектуры (в компьютере, за которым я пишу этот текст, как раз один из таких), и их официальное название было `EMT64`, а затем `Intel 64`; все на тот момент уже пользовались названием `AMD64`, и оно закрепилось. Независимое от производителя название — `x86_64`.

Если вы выбрали нужный пакет и ошибка все еще возникает, проверьте, что любые зависимости, которые вы могли установить для этого пакета,

та, также имеют верную архитектуру. Если больше ничего не помогает, на DVD есть и исходный код – попробуйте установить программу из него с помощью инструкций в tar-архиве, это гарантирует вам компиляцию для нужной архитектуры.

6 Трата свободного места

В Я обновил жесткий диск своего ноутбука с 40 ГБ до 250 ГБ. Исходный диск в 40 ГБ (/dev/sda1 и раздел подкачки /dev/sda5) я клонировал в диск 250 ГБ с помощью CloneZilla, и после установки нового диска с него прекрасно загрузилась Ubuntu 9.04.

Я ожидал, что CloneZilla увеличит размеры разделов чтобы использовать новое место. К сожалению, новые разделы того же размера – 40 ГБ с неиспользуемой областью в 201 ГБ. Затем я загрузил GParted с Live CD, но в ней можно было только уменьшить размеры разделов, так как неиспользуемое место не отображалось. Как увеличить размер разделов?

Стюарт Мак-Федьен [Stuart McFadyen]

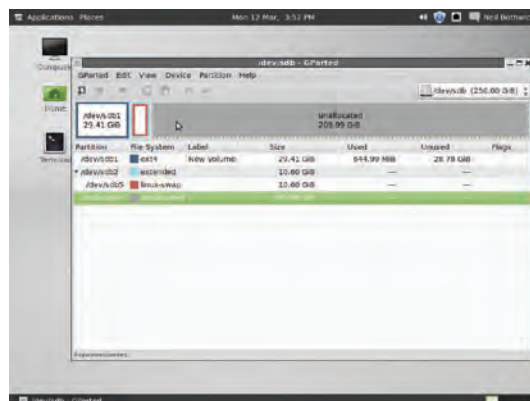
Вы столкнулись с ограничением таблицы разделов MS DOS, используемой уже 30 лет с гаком. Это ограничение в четыре первичных раздела (которое было излишним во времена жестких дисков размерами в 10 МБ). Чтобы можно было добавить дополнительные разделы, один из них нужно сделать расширенным, и он сможет включать в себя логические разделы.

Ваш загрузочный раздел – первичный. Хотя раздел подкачки **swap** находится на логическом разделе, они нумеруются от 5 и выше. Клонировав диск, вы создали копию разделов жесткого диска в 40 ГБ, и еще 200 ГБ свободно в конце. На них можно создавать добавочные разделы и монтировать их куда требуется (например, организовать отдельный раздел для **/home**), либо переместить раздел **swap** в конец диска, чтобы можно было менять размер **sda1**. Раздел подкачки **swap** можно переместить и в работающей системе, но проще воспользоваться Live CD.

Загрузитесь с Live CD, запустите GParted (или QtParted в Live CD с KDE), удалите раздел **swap** и содержащий его расширенный раздел (скорее всего, **sda2**, если на диске больше ничего нет). Затем создайте новый раздел **swap** в конце диска, чтобы неиспользуемое пространство находилось сразу после **sda1**. Теперь измените размер **sda1**. Потом смонтируйте **sda1** и измените **etc/fstab** для этого раздела. Измените в строке **swap** ранее используемый UUID на имя устройства (типа **/dev/sda2**) или на новый UUID. Чтобы его узнать, откройте терминал и запустите команду

```
sudo blkid /dev/sda2
```

Решив сделать это, не прекращая работы системы, сначала отключите раздел **swap**:



▶ Запутанная таблица разделов MS DOS усложняет перераспределение диска на разделы.

```
sudo swapoff /dev/sda5
```

Затем запустите GParted, измените **fstab** и снова включите **swap** командой

```
sudo swapon /dev/sda5
```

Далее при первой загрузке нажмите **e** в меню **Grub** для изменения параметров загрузки, удалите **quiet splash** из строки конфигурации ядра и нажмите **Ctrl+X**, чтобы продолжить загрузку. Так мы выключим заставку только для этой загрузки, чтобы увидеть сообщения об ошибках.

Работу **swap** после загрузки можно проверить командой

```
cat /proc/swaps
```

LXF



Часто задаваемые вопросы

Волшебные кнопки

▶ **Зачем компьютерам кнопка перезагрузки?**

Потому что они поставляются с операционной системой от Microsoft.

▶ **Эта шутка устарела еще при динозаврах. По-вашему, в Linux программы никогда не падают?**

Ядро Linux весьма стабильно, и падение программы обычно не слишком затрагивает остальную часть системы. Но иногда все может испортить неконтролируемый процесс, захвативший все ресурсы процессора или памяти (или и те, и другие), и может потребоваться перезагрузка. Или паника ядра может остановить едва начавшийся процесс загрузки.

▶ **Значит, нам все-таки нужна эта ужасная кнопка сброса?**

Нет, не нужна. В ядре Linux предусмотрено маленькое волшебство.

Если нажать **Alt**, **SysRq** и командную клавишу, ядро ответит на нее, если оно не заблокировано полностью. Так как это функция ядра, она начинает работать после загрузки ядра, и ею можно пользоваться в процессе загрузки системы.

▶ **А у меня нет кнопки SysRq – тогда что делать?**

Клавиша **SysRq** также используется для печати с экрана (**Print Screen**, **PrtSc**) – на ней может быть любое название из этих двух или оба одновременно.

▶ **Что такое командные клавиши?**

Есть несколько командных клавиш для выполнения различных операций, но самые полезные – **R**, **E**, **I**, **S**, **U** и **B**, нажатые в этом порядке. **R** сбрасывает драйвер клавиатуры в «сырой» режим, **E** отправляет всем процессам сигнал **TERM**, **I** отправляет

сигнал **KILL**. Не используйте две последних клавиши при удаленном подключении, потому что они прервут сетевое соединение. **S** синхронизирует все смонтированные файловые системы, то есть немедленно записывает все закэшированные данные на диск. **U** размонтирует все смонтированные файловые системы, затем перемонтирует их в режиме только для чтения. Последние две операции гарантируют, что содержимое дисков будет корректно записано и не сможет быть повреждено. После этого можно безопасно нажать третью командную клавишу **B** для немедленной перезагрузки системы.

Так как все диски синхронизированы, то вы не увидите предупреждений после перезагрузки, и будет не нужно запускать **fsck**.

▶ **Как запомнить эту последовательность?**

Есть несколько расшифровок; некоторые очень глупые. Можете воспользоваться таким – «Reboot

Even If System Utterly Broken [перезагрузиться, даже если система совсем вышла из строя]», что весьма точно передает суть, или запомнить, что это «**BUSIER**», читаемое задом наперед.

▶ **Все это хорошо, если у компьютера есть клавиатура; ну, а если это удаленный или автономный компьютер?**

В этом случае запустите следующую волшебную команду:

```
echo s >/proc/sysrq-trigger
```

Если вы можете подключить к обложке компьютера (**shell**), то сможете и синхронизировать диски (**sync**), и завершить работу системы (**shutdown**).

▶ **Существуют ли другие полезные клавиши и команды?**

Полный список и другие важные подробности можно найти в документации ядра в файле **/usr/src/linux/Documentation/sysrq.txt** или на сайте www.mjmwired.net/kernel/Documentation/sysrq.txt.

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.

MyJgui » Urban Lightscape » SysUsage » Din » Mundus » MP3Diags » kenozoid » Avoision » Jskat » Mutiara » Bluefish

Менеджер баз данных

MyJgui

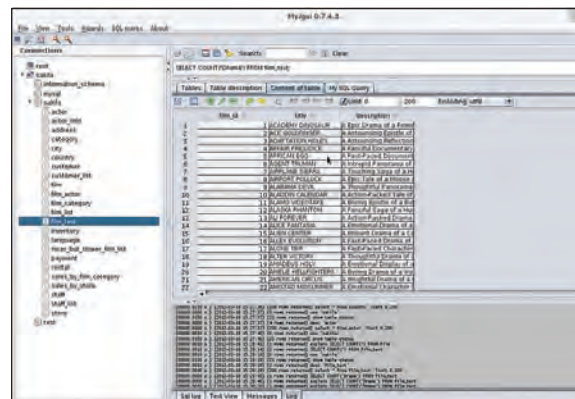
Версия 0.7.4.3 Сайт <http://myjgui.com>

Базы данных являются такой же неотъемлемой частью нашей жизни, как налоги, грозовые тучи и жалобы на мои шуточки. Тут я должен извиниться перед администраторами баз данных. Базы данных чрезвычайно милы, и мы все их искренне любим. Но большинству они хоть и нужны, но не особо интересны. Конечно, такие люди не горят желанием тратить время, проделывая

со своими таблицами потенциально опасные штуки, чтобы разобраться в оболочке.

Долгое время главным инструментом был *phpmyadmin* – сетевой инструмент, с легкостью справлявшийся со всеми видами стандартных задач. *MyJgui* использует иной подход. Это написанное на Java приложение рабочего стола, которое в большей или меньшей степени делает все то же, что и *phpmyadmin* – но, наверное, немного лучше?

Ну, об этом каждый судит сам – хотя освобождение от тирании окна браузера уже яв-



» Базы данных могут быть сущей конфеткой! Кхе... ну, по крайней мере, они могут быть не такими уж противными.

«Освобождение от окна браузера – уже отличный стартовый пункт.»

но отличный стартовый пункт. И он лучше, это правда. Навигация по вашей базе данных не требует бесконечных обновлений экрана для одной задачи – в левой стороне находится симпатичное древовидное отображение. Вы даже можете управлять соединениями со множеством разных хостов, не ограничиваясь одним сервером.

Правая область разделена на три панели, и предоставляет вам окно Query [Запрос], браузер таблиц Table [Таблица] и окно Output [Результат]. Два последних имеют вкладки, и легко переключиться, например, между таблицей, полученной в результате вашего запроса, и списком всех таблиц. Подобным же образом окно Output может отображать сообщения, журнал MySQL или любой расширенный текст из выделенного поля.

Однако этот отличный инструмент не сделает все за вас – очевидно, вам все же придется уметь писать запросы SQL, чтобы выполнить работу, но для простых задач, таких, как создание таблиц и ввод данных, он достаточно прост, чтобы в нем разобрался новичок.

Будучи приложением Java, оно, естественно, запускается прямо из файла jar: `java -jar MyJgui.jar` и у вас не будет никаких проблем с зависимостями. И внешность у него тоже приятная.

Исследуем MyJgui

Соединение

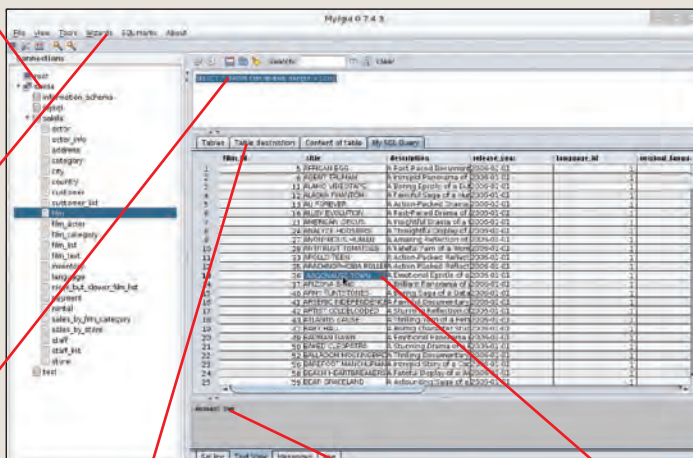
Управляйте соединениями с разными серверами.

Работает, как по волшебству

Мастера программы помогают создавать базы данных и таблицы.

Команды SQL

Здесь вводятся команды SQL для работы.



Отображение во вкладках

Отображение во вкладках [Tabbed view] показывает результаты или данные таблиц.

Щелкайте и редактируйте

Двойной щелчок по записи позволяет ее отредактировать.

Результат

Панель Output отображает отклик от движка базы данных.

Графическая утилита

Urban Lightscape

Версия 1.0.4 Сайт <http://bit.ly/zAhRh1>

Фотografiрование – плевое дело: знай наводи свой Hasselblad/Fuji/Canon/телефон, нажми на кнопку, и готово. Однако делать хорошие фотографии несколько сложнее. И даже если вы отлично разместили все в кадре и не забыли о композиции и фокусе, всегда остается освещение.

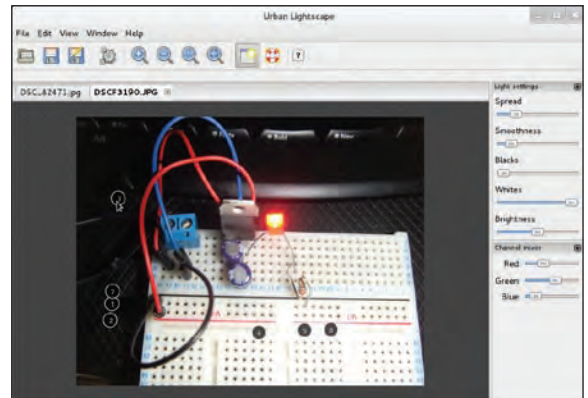
В фотографии освещение – это все, но заставить камеру увидеть изображение так же, как умеют ваши глаза, непросто. Задача, потому-то в студийной фотографии применяется столько осветительных приборов и измерений. Есть способы поправить изображение так, чтобы выявить находящиеся в тени детали, не «раздувая» яркости, но один из самых простых – употребить *Urban Lightscape*.

Загрузите изображение, и оно отобразится в окне, как и следовало ожидать, снабженном боковой панелью с ползунками для настройки всяких параметров. И вот где сокрыто волшебство – щелкните

дважды в области изображения, и оно превратится в Контрольную точку [Control point]. Щелкните дважды по этой точке и удерживайте ее, а потом перемещайте мышью вверх и вниз, чтобы настроить относительную яркость этой точки!

Очевидно, лучше всего это работает, если у вас есть много Контрольных точек по всему изображению, в местах, требующих настройки. Ползунок Smoothness справа управляет интерполяцией освещения между этими точками. Разумная настройка этих точек (щелкните по ним и перетаскивайте, чтобы их переместить) и ползунки помогут выявить скрытые детали практически любого изображения. Даже сложное изображение можно таким

«Даже сложное изображение можно сильно усовершенствовать.»



► **Сделайте темные места светлее, а светлые темнее. Сделать проще, чем сказать — с помощью *Urban Lightscape*.**

образом сильно усовершенствовать за относительно малый промежуток времени. Конечно, умнее было бы выбрать должное освещение в момент съемки, но для «вытягивания» фотографий, отснять которые заново уже невозможно, лучше *Urban Lightscape* опции не найти.

Полный исходник находится на DVD. Помимо Wxwidgets, используемых для GUI, здесь нет никаких экстраординарных зависимостей, так что компиляция и установка должны протечь со свистом.

Монитор системы

SysUsage

Версия 5.2 Сайт <http://sysusage.darold.net>

Когда вы спите и видите сладкий сон о своей личной уставленной мониторами штаб-квартире, что вам представляется на экранах? На одном-двух из них уж обязательно должно быть крутое отображение извилистых кривых.

SysUsage не только предоставляет подобный живописный фон, но также приносит немалую пользу. Загогулины, выводимые им на экран – это детальное описание статуса вашей системы, те вещи, которые вы должны знать: например, использование CPU, пропускная способность сети, свободное место на диске и прочее. Знание всего этого очень важно, если у вас имеется удаленный сервер, вот почему *SysUsage* является сетевым и способен работать в качестве интерфейса для различных отчетов системы.

За сценой *SysUsage* активно использует замечательный инструмент *RRDTool*. Это крайне полезный фрагмент кода, который довольно профессионально занима-

ется всем, что касается сбора данных – или сбором и отображением «данных временного ряда», как любят говорить профи в этой сфере. Это означает, что за *SysUsage* стоит солидная и крепкая база, которую он «всего лишь» оборачивает в удобный интерфейс.

SysUsage также обеспечивает поддержку модулей расширения, так что можно без проблем подключить сервисы и их статистику, подлежащую мониторингу. Нужды *MySQL* и *Nginx* (альтернативный веб-сервер) переключает не так уж много инструментов, но, по крайней мере, имеется опция, позволяющая запустить руки в код, если вы захотите расширить его самостоятельно. Установка *SysUsage* довольно проста. Он написан на Perl, и мо-

«За сценой SysUsage использует замечательный RRDTool.»

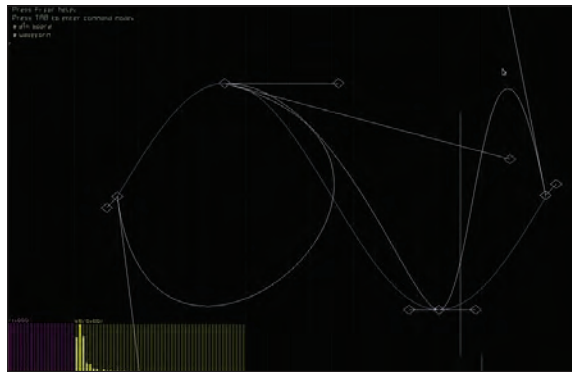


► **Теперь, с *SysUsage*, ваша злобная штаб-квартира будет выглядеть должным образом. А вы сможете проверять свою серверную ферму.**

жет потребовать, чтобы вы скачали дополнительные модули с CPAN. Помимо этого, основная зависимость – *rrdtool*, который имеется в репозиториях многих дистрибутивов. Однако желательно все же прочесть онлайн-документацию по установке и настройке – не то чтобы они были очень трудны, но там есть кое-какие хитрости и опции, которые могут вам пригодиться, например, интеграция с *Nagios* или распределенные отчеты.

Аудиопрограмма

Din

Версия 2.0 Сайт www.dinisnoise.org

► Управляйте собственными звуковыми пейзажами с помощью уникальных звуковых кривых.

Мы уже рассматривали *din* (*din is noise* – это рекурсивный акроним) на священных страницах HotPicks. Но то был старый *din*. А этот – сильно измененная сущность, дошедшая уже до блистательного номера версии 2.0.

Но сначала – небольшое отступление. Создание музыки – это не только умение играть на гитаре или клавишных. Для многих возможность извлечь звук из электронного эфира является более быстрым и очевидным способом атаки на барабанные перепонки. И *din* предоставляет вам для этого нужный инструмент.

Выбор из меню звуков или даже манипуляцию основными волнами предоставляют многие программы, но немного найдется таких (нам в голову приходит только одна), которые позволяют вам самим рисовать волну.

И мы тут подразумеваем не кое-как нацарапанные синусоиды, а конструкцию из хорошо определенных и редактируемых кривых Безье.

Din делает замечательную вещь, заинтересовывая вас игрой со звуками: удовольствием становится сам акт творения, а не только прослушивание результата (ваше счастье, что вы не слышали некоторых наших композиций). Хотя, конечно, если вы хотите создавать музыку для пользы и в назидание другим, это тоже возможно.

Хитрость использования *din* в том, чтобы прочитать инструкции. Или, в самом крайнем случае, встроенную систему подсказки. По большей части, что вполне естественно, управление осуществляется с клавиатуры, либо вводом текстовых команд, либо с помощью горячих клавиш, и, честно говоря, многие экраны весьма похожи друг на друга, так что не худо бы-

«Din заинтересовывает вас игрой со звуками.»

ло бы познакомиться с каждым, прежде чем пытаться сделать что-то еще.

Никаких ужасно неожиданных зависимостей в *din* нет. Он требует наличия рабочего и доступного звукового сервера *Jack*, что в одних дистрибутивах труднее, в других проще (запустить-то просто, а вот сделать управляемым пользователем сложнее), но в любом случае стоит приложить усилие и заглянуть в документацию и для *din*, и для самого *Jack*.

Автору, создавшему прекрасную музыку, хотелось бы услышать ее живьем – и нам тоже. Почему бы вам не разместить ссылку на ваши чудные опусы на форумах *Linux Format*?

Инструмент для наведения порядка

Mundus

Версия 1.1.3 Сайт <http://mundus.sebikul.com>

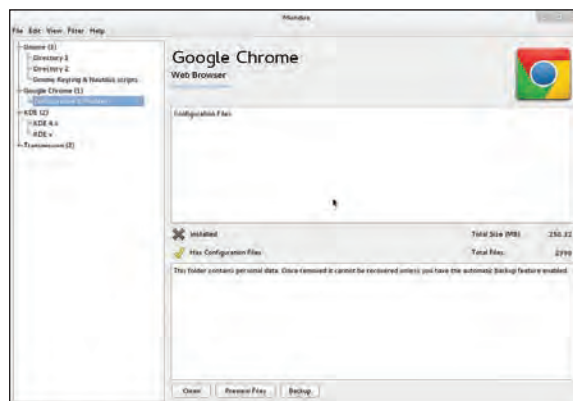
При повседневном использовании настольного Linux возникают хронические неудобства. Мы имеем в виду не зигзаги парадигмы рабочего стола, а всего лишь накопление мусора в домашней папке. Если в других ОС есть тенденция отгрести этот мусор с дороги, то в Linux он видим, и если вам нужно найти определенный файл в *~/*, придется продирааться сквозь множество записей и директорий, созданных приложениями, которыми вы средо не пользовались.

Mundus (не путать с дистрибутивом с тем же названием) – небольшая программа, которая пытается облегчить ваше бремя. Запустив ее, вы увидите экран с панелями, со списком распознанных установленных программ в левой стороне, а панель информации находится справа. Цифра рядом с названием программы означает количество вхождений. *Mundus* распознает не все – в нем есть список программ, с которыми он согласен рабо-

тать. В зависимости от программы, у вас есть ряд опций, таких, как очистка кэша или удаление файлов настройки. В каждом случае вы можете также сделать резервные копии установленных данных. Это не супер-надежные копии, потому что они хранятся в директории *Mundus* в домашней папке, но вы можете скопировать их еще куда-нибудь.

Программа не лишена проблем, так как она экспериментальная. Например, при использовании фильтра для отображения вхождений только программ, которые не были установлены, было странно увидеть в появившемся списке *Chrome*, *KDE* и *Gnome*. Эта программа может весьма пригодиться, но не стоит делать слиш-

«Программа не лишена проблем, так как она экспериментальная.»



► Приборка – не самое приятное занятие, но есть способы сделать ее проще.

ком высокие ставки на то, что она всегда будет работать корректно. *Mundus* написан на *gambas*, и требует корректной работы ряда модулей *gambas*, причем не все они устанавливаются при изначальной попытке ее компиляции; так что если она никак не хочет у вас работать, просто установите наиболее вероятные модули (или, на самом деле, вообще все подряд, потому что, честно говоря, не так уж их много).

Аудиоинструмент

MP3Diags

Версия 1.1.17 Сайт <http://mp3diags.sourceforge.net>

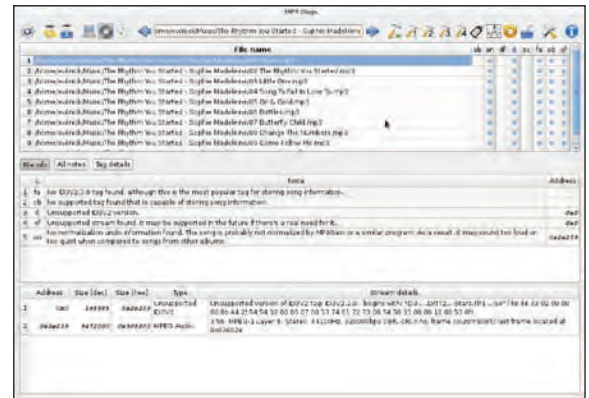
Клучшему оно или к худшему, но разменной монетой в мире музыки ныне является файл MP3. С этим форматом куча проблем, и если бы сегодня такое создавали бы с нуля, то результат был бы иным. Одна из проблем на самом деле связана не с самим форматом, а с добавленными к нему украшениями – тэгами ID3.

Тэгирующие музыкальных файлов подобным образом отнюдь не является главной задачей, если только вы не хотите лишиться рассудка, и тем не менее существует масса вариаций формата тэгов ID3. Worse того, они используются самыми различными способами, и, к вящей досаде, еще и вразноык. Даже у «профессиональных» файлов MP3, которые продаются на Amazon в противопоставление вашему самоделькам, имеются всевозможные неувязки.

MP3Diags – инструмент, который усердно процеживает вашу музыкальную коллекцию в поисках проблем. Это могут

быть проблемы с самой аудиозаписью (испорченные или неверно заверненные блоки могут вообще не позволить воспроизводить файл, в частности, на портативном оборудовании) или с информацией, встроенной в нее. Инструменты поиска настраиваемы, так что вы можете проверять, например, файлы со встроенными иллюстрациями. Также вы можете соединиться с онлайн-сервисами и попытаться идентифицировать и сгенерировать пропущенные данные тэгов.

MP3Diags уже рассматривался в HotPicks, но в этот новый релиз включен ряд отличных функций, таких, как режим командной строки для некоторой автоматизации работы и дополнительные опции для автоматического удаления или



► Тэги ID3 содержат много полезной информации. А также бесполезной. И часто вообще неверной.

добавления определенных типов данных. Нельзя не отметить, что к пользовательскому интерфейсу по-прежнему надо какое-то время привыкать, но оно того стоит – на данный момент нет лучшего способа оздоровить большую коллекцию музыки с меньшей морокой.

Естественно, для работы с MP3Diags вам понадобится наличие соответствующих библиотек MP3. И стоит отметить (хотя мы никогда не сталкивались с какими-либо проблемами с этой программой), что она переписывает ваши исходные файлы, так что будет разумно сделать резервные копии всего, что вы сочтете важным.

«Нет лучшего способа оздоровить большую коллекцию музыки.»

Программа для дайвинга

Kenozooid

Версия 0.9.0 Сайт <http://bit.ly/HdiUJO>

Эта программа взята с полки «для узкоспециального использования», но нам в HotPicks не зачем за это извиняться. По сути, специализированные программы принадлежат именно к тем областям, для которых и хорош открытый код, поскольку обычно это подразумевает малую целевую аудиторию и, следовательно, «ограниченную коммерческую ценность».

Kenozooid создан специально для любителей дайвинга – в смысле, настоящего дайвинга, то есть под водой и все такое.

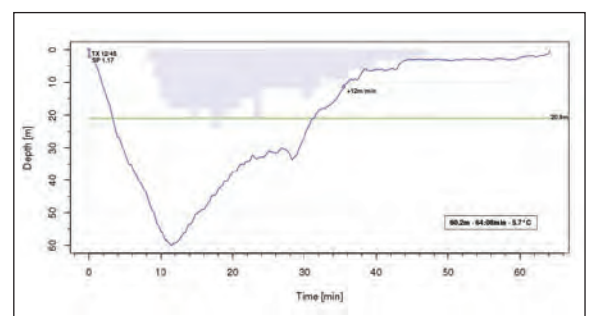
Одна из главных забот дайверов обычно в том, сколько они могут пробыть под водой. Тут много разных сложных факторов – например, как долго можно оставаться на определенной глубине; это очень трудно рассчитать, находясь в погружении.

Kenozooid не только поможет рассчитать ваш нырок (т.е. спланировать время вашего пребывания на разных глубинах),

но, если у вас есть устройство подводной записи/компьютер, также извлечет немалую пользу из записанных вами данных. Он хранит данные в формате Universal Dive Data Format (UDDF), поэтому он совместим и может взаимодействовать с другими устройствами и программами.

Да, на такие вещи действительно есть стандарт, и он используется практически всем соответствующим оборудованием и программами (его придерживаются и люди!). Kenozooid написан на Python3. Да! Так что когда вы запустите скрипт установки, проверьте, что вызывается именно Python3, а не «обычный» Python, который до сих пор используется по умолчанию в большинстве ОС:

«Kenozooid извлечет пользу из записанных вами данных.»



► Мы не сплещу в дайвинге. Однако, судя по виду, сплещам это может пригодиться.

```
sudo python3 setup.py install
```

Он сам позаботится об установке и обеспечит все необходимые зависимости.

Иногда вы будете наталкиваться на пакеты, не распознаваемые в Python3 – это часто происходит по причине легкого несовпадения названий, потому что способов обозначения совместимости с Python3 очень много; так что вам, возможно, придется мобилизовать умственные способности.

HotGames Развлекательные приложения

Минималистская игра

Avoision

Версия 1.1 Сайт <http://avsn.sourceforge.net>

История компьютерных игр знавала ужасные примеры. Многие из ужасных примеров отличались мощными командами разработчиков, затейливыми графическими наворотами, профессионально записанными саундтреками... и прискорбной недалёковидностью. Эта игра не из таковых. Здесь нет затейливой графики, саундтреков от Джона Уильямса [John Williams] или плеяды именитых разработчиков; нет даже сюжета или того, о чем можно было бы поговорить. Зато есть минималистское искусство в чистом виде: игра сведена исключительно к голым основам, до такой степени, что начинаешь задаваться вопросом: «А это вообще игра?»

Мы думаем, что да. Несложная, это верно, но в ней есть существенный элемент игрового процесса. Цель игры проста: управляя красным кубиком, надо

двигать его к зеленому кубу (используя стрелки). И это все. Правда, есть еще белые кубики. Их много, и вам будет плохо, если они в вас врежутся. Ну, то есть, вы проиграете, а это всегда нехорошо. Белые двигаются очень быстро, и вам понадобится высокая подвижность, а иногда, возможно, и ум, чтобы от них ускользнуть.

Если вам это поможет, представьте себе, что вы играете в официальную лицензию *The Walking Dead*, и белые блоки – это зомби, а зеленые блоки – ваш трейлер. Только тут нет пистолетов. Или хрюкающих звуков. А сюжет практически тот же. Если хотите усложнить себе жизнь, здесь

«Управляя красным кубиком, надо двигать его к зеленому.»



» Уворачивайтесь от жутких 3D-зомби, когда они будут пытаться... короче, просто старайтесь, чтобы в красный кубик не попали.

есть различные уровни, включая экстремальный режим [Extreme mode]. Как далеко вы зайдете?

К счастью, отсутствие обычной накипи означает, что у этой игры чрезвычайно мало зависимостей, и она скомпилируется практически на любом компьютере, на котором запускается SDL. Просто следуйте обычной процедуре `configure/compile/make` и готовьтесь уворачиваться.

Карточная игра

JSkat

Версия 1.10.1 Сайт www.jskat.org

Скат, как вы можете знать или не знать, это карточная игра исключительно для трех игроков. Впрочем, в век искусственного интеллекта [AI] уже не обязательно принуждать дедушку или ни на что другое не годного кузена составить вам компанию. Обладание компьютером дает вам определенную свободу.

Игра, придуманная в Германии, имеет немало хитрых вариаций, но основное ее правило – надо выигрывать взятки, как в висте или бридже. После сдачи игроки «торгуются», озвучивая, сколько очков они намерены набрать, а победитель в торговле должен попытаться добыть заявленное количество очков за десять раундов. Два других игрока – понятное дело, несколько расстроенные торговым невезением – сговариваются помешать «заявителю» достичь своей цели.

Возможно, это звучит сложнее, чем есть на самом деле, но, к счастью, дружелюбная версия этой игры, *JSkat*, легко введет вас в курс дела. Как уже упоминалось, есть несколько версий, и бесчисленное множество дополнительных правил. Хотя *JSkat* не включает абсолютно все дополнения, в нем есть основные типы игр и некоторые популярные варианты; планируется также ввести и большее.

С точки зрения игрового процесса, здесь есть три доступных типа игроков AI, с разными тактиками. Если вы заскучали от их стиля игры и решили внести свой вклад в код, здесь есть API, чтобы

«ПО JSkat красочно, компетентно и просто В ИСПОЛЬЗОВАНИИ.»



» Это очень важный экранный снимок: мы в первый раз победили в торговле!

вы могли написать свой собственный AI. Да, и вы можете играть онлайн. Скат – настолько популярная игра, что у нее есть свой Международный сервер (International Skat Server), который соединит вас с другими игроками, независимо от того, какое ПО они используют. По большей части это немцы, потому что скат там особо популярен.

Программное обеспечение *JSkat* достаточно красочно, компетентно и, что важно, просто в использовании. Так как здесь всего один .jar-файл Java, нет нужды что-либо устанавливать или возиться с компиляцией – просто запустите его и начните играть.

Графическая программа

Mutiara

Версия 0.9.2 Сайт <http://mutiara.sourceforge.net>

Человек рисует мозгами, а не руками. Это, между прочим, высказывание Микеланджело, так что все претензии направляйте в XVI век. Высказывание гения, безусловно, справедливо, но беда в том, что у многих, похоже, руки не способны реализовать великие замыслы мозгов.

С целью помочь этому горю и разрабатываются инструменты вроде *Mutiara*. Это не просто пакет для творчества, как *GIMP*. И даже не пакет для рисования, как *Inkscape*, хотя это уже теплее. *Mutiara* (я тоже понятия не имею, что сие слово означает) является создателем узоров – красивых узоров по шаблонам рекурсивных алгоритмов рисования.

Некоторые варианты включают мотивы для логотипов или web-страниц, и возможность создавать на экране эти фрактало-подобные объекты и растворяться в глубокой и таинственной красоте визуализированной математики буквально заворачивает. Оплошностью программы является преступное отсутствие инструкций

или руководств. Есть незначительные подробности в диалоговом окне About > Help, они помогут вам немного сориентироваться в основах.

Программа работает в двух режимах: рисование с помощью мотивов и создание самих мотивов. Как использовать вторую часть, понятно – щелкните по кнопке Motif под заголовком Replication, чтобы перейти в редактор. Повозитесь с инструментами рисования и выберите Apply Motif для сохранения своего творения, а затем вы сможете использовать его наравне с другими предузавренными образцами.



› Это как вышивка, только с помощью фракталов и компьютера вместо иглолки.

Web-редактор

Bluefish

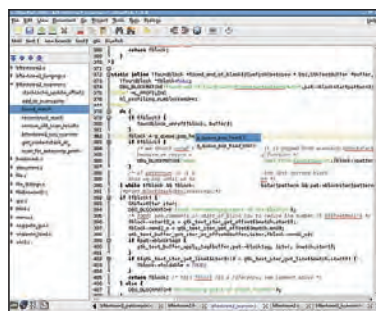
Версия 2.2 Сайт <http://bit.ly/xld3>

Разве кто-нибудь еще пользуется web-редакторами? Раньше таких программ были дюжины, но уцелела всего пара активных проектов. Неужели *Bluefish* настолько хорош? Или, возможно, больше никто не пользуется Интернетом...

Bluefish имеет долгую историю, и о нем уже говорилось в HotPicks. Но всегда есть что добавить. У него уже было впечатляющее число функций редактирования для работы с разными видами HTML и CSS, а также способность генерировать скрипты и делать всякие умные вещи с раскладкой.

Недавние релизы включили также поддержку кодирования zen, что сокращает долгое и сложное занятие по вложению множества общих элементов. Предыдущий релиз, как раз перед Рождеством, ввел эту функцию, но было множество проблем с некорректной работой курсора, и вам будет приятно узнать, что сейчас все отлажено и работает нормально.

Есть другие способы управления web-разработкой, но *Bluefish* блистает функциями и умениями. Приверженцы старой школы могут предпочесть создавать все вложенные скобки вручную в *Vim* или *Emacs*, но для 99 % остальных *Bluefish* предлагает полезные и экономящие время функции и способность тестировать и отлаживать материал для размещения в сети безопасно и относительно спокойно. LXF



› *Bluefish* — теперь он идет с добавленным zen.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

› BuildBot 0.8.6

Автоматизируйте процесс сборки посредством компиляции и тестирования после каждого внесенного в код изменения.

<http://trac.buildbot.net>

› ClamTk 4.38

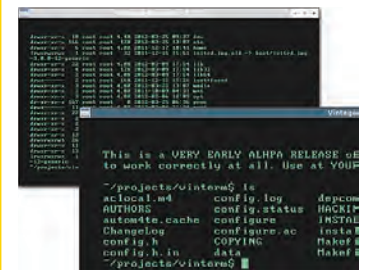
В дружелюбном интерфейсе защитника от вирусов наведен порядок.

<http://clamtk.sourceforge.net>

› Vintage Terminal 0.1.0

Верните свою консоль в восьмидесятые с помощью этой превосходной ретро-CLI.

<http://code.google.com/p/vinterm>



› «Партику в шахматы?»

› LDAP Account Manager 3.7

Это нудно, но это приходится делать – сортируйте людей самым простым способом.

www.ldap-account-manager.org

› Berusky 1.5

Заново изобретенный Sokoban, с отличной графикой и новыми головоломками.

<http://anakreon.cz/?q=node/1>

› Mathematic 15.8.2

Отличный обучающий математический пакет, который понимает формулы.

www.mathomatic.org/math

› PartedMagic 2012_3_24

Самый полный инструмент для восстановления/резервного копирования диска.

<http://partedmagic.com/doku.php>



› Удобное копирование данных.

На диске

Попробуйте новую операционную систему сегодня!



В этом месяце у нас необычный DVD, не в последнюю очередь из-за того, что большинство операционных систем на нем не относятся к Linux. Это вызывает небольшую проблему, поскольку наш метод для live-загрузки не всегда ладит с другими ОС. Поэтому мы включили на диск файлы ISO-или USB-образов. Желая загрузить систему, вы можете либо

перенести такой файл на физический носитель, либо, в качестве альтернативы, мы включили удобный скрипт, позволяющий запустить виртуальную машину в *Qemu*, где ОС будет работать.

Но мы все-таки *Linux Format*, а не *Все-разные-ОС Format*, поэтому включили Tails — дистрибутив Linux, созданный специально для обеспечения вашей конфиденциальности онлайн.

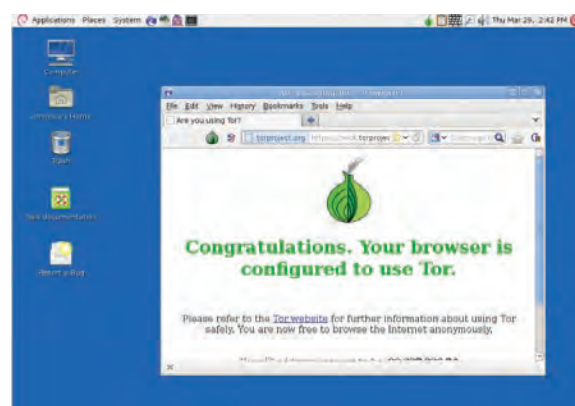
Debian со сверхзащитой

Tails

Есть множество инструментов, помогающих защитить конфиденциальность онлайн, но не всегда легко узнать, правильно ли вы их применяете. Например, вы, возможно, используете Tor, но вы (или тот, кто влез на ваш компьютер) установили модуль, выдающий вас. Есть и вероятность попасть на скомпрометированный компьютер — не секрет, что компании и интернет-кафе используют на своих машинах фальшивые сертификаты, чтобы они могли читать контент HTTPS. Цель Tails — решить эти проблемы, предоставив вам настроенный и готовый к ра-

боте дистрибутив live, обладающий всей безопасностью и конфиденциальностью открытого кода. Просто загрузите DVD, и вы получите безопасную среду.

По умолчанию все будет заблокировано, а ваш трафик — зашифрован и анонимен, и вы не наследите на компьютере, работая с Tails. Настроенная по умолчанию безопасность не дуракоупорна: то, что включено, можно включить — например, cookies и скрипты сайтов. Тогда вы потенциально раскроетесь любителям шпионажа. Чтобы повысить свою безопасность, стоит перезагружаться всякий раз, когда



► Tails использует Tor, чтобы обеспечить конфиденциальность работы в сети.

вы хотите использовать другую онлайн-личину, например, другой адрес электронной почты или учетную запись в чате. Система не имеет опции установки — это сделало бы ее открытой для вторжения.

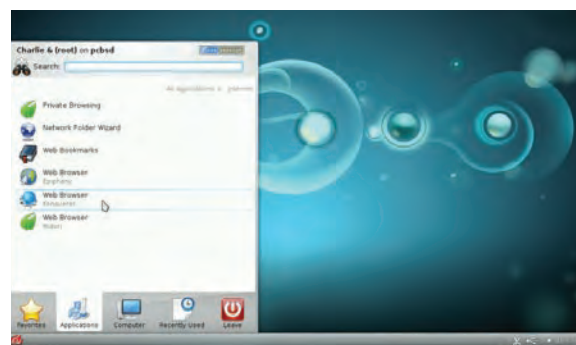
Легендарная прочность и удобный интерфейс

PC-BSD

Из-за ограниченности объема DVD мы смогли включить только одну из Unix-подобных операционных систем из Сравнения. Мы остановились на PC-BSD из-за числа поддерживаемых систем и программ. Чтобы запустить эту систему, найдите окно в *Bootimg.sh*. Попад в меню загрузки PC-BSD, щелкните внутри окна, чтобы ввести данные с клавиатуры, выберите 6, чтобы обеспечить корректную графику, а затем нажмите на Enter.

Если раньше вы не использовали BSD, вас, возможно, удивит ее знакомый вид. Собственно, на первый взгляд ее трудно отличить от дистрибутива Linux. Это потому, что самое большое различие между ними, ядро, скрыто под слоями общих для них программ. Первым вы увидите менеджер входа, который предложит вам выбрать между KDE, Gnome, LXDE и Xfce.

На самом рабочем столе множество программ, которые приличествуют дист-



► BSD — это не только для серверов.

рибутиву Linux. При запуске из скрипта *Bootimg.sh* рабочий стол может показаться немного медлительным — но дело в том, что он работает в виртуальной машине.

Установка одним щелчком

Ubuntu и OpenSUSE уже освоили установку одним щелчком, и теперь мы их догнали! На странице DVD HTML есть ссылки, автоматически устанавливающие программы (там, где это поддерживается) — просто щелкните по ссылке для своего дистрибутива.

Пакеты берутся из главного репозитория, и понадобится интернет-соединение. Правда, один щелчок — это слегка оптимистично (нам пришлось утрудить мышь чуть больше), и нужен пароль, но для новичков это явно отличный метод установки программ.

Если вы — любитель работы в браузере, можете установить все программы таким образом, перейдя в www.appnr.com (Ubuntu) или software.opensuse.org. А нет выхода в Сеть — не беда: наш диск по-прежнему будет включать пакеты и исходный код.

В BeOS вдохнули новую жизнь

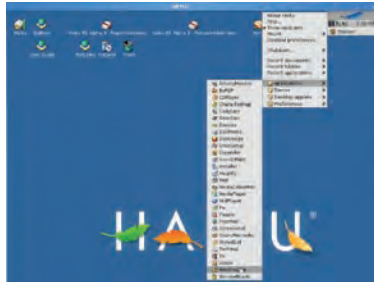
Haiku

Помните BeOS? Возможно, где-то в глубине вашей памяти мелькнула смутная искра узнавания. Что ж, позвольте освежить вашу память. BeOS была операционной системой, разработанной для пользователей мультимедиа; вышла она в конце девяностых, а просуществовала до 2001 года. По причинам скорее коммерческим, нежели техническим, она никогда не была особо популярна, зато привлекла к себе хоть и немногих, но преданных последователей. Эти люди создали Haiku, свободную систему, основанную на BeOS.

Только истинные фанаты BeOS будут использовать эту систему постоянно, но это не значит, что она не стоит внимания. В отличие от Linux и прочих Unix-подобных систем, Haiku разработана по большей части для пользователей рабочего стола – в частности, для пользователей мультиме-

диа рабочего стола. И ядро, и файловая система, и прочие аспекты системы созданы в соответствии с этой концепцией.

Загляните в документацию **Welcome** и **User Guide на рабочем столе**, или щелкните по перышку в верхнем правом углу, чтобы вывести меню Applications, где найдется все необходимое для начала.



➤ Рабочий стол Haiku: поэзия в движении.

Важно ВНИМАНИЕ!

➤ **Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.**

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru, для получения содействия.

Открытый код берется за AmigaOS

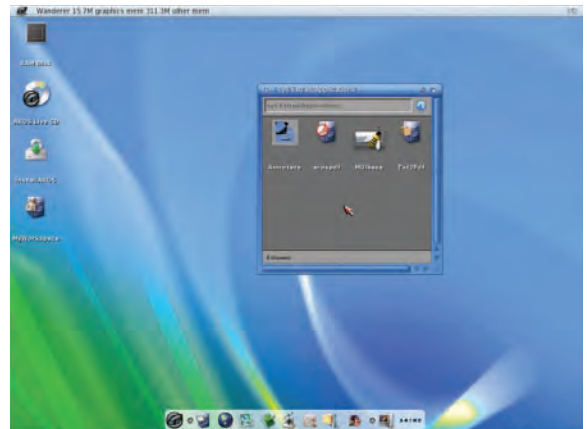
Icaros

Возможно, мы видим прошлое в розовом свете, но Amiga – это было здорово. В ее времена ожидание загрузки компьютера было волнующим, а не досадным. Готовя этот DVD, мы с удивлением узнали, что AmigaOS продолжает разработку, и последняя версия (4.1u4) вышла в декабре 2011. Однако код ее закрыт, и она проприетарная.

Но ностальгирующие фанаты все же могут получить свою дозу Amiga-видности

с помощью Icaros. Это дистрибутив Aros, являющий собой ОС, основанную на AmigaOS API. Важно отметить, что это не клон AmigaOS, а реализация ее API в новой ОС.

Щелкните по значку с глазом на панели для вывода меню Applications, или по изображению папок, чтобы перейти в подборку руководств. В ней больше приложений, чем в Haiku, и среди них есть готовые к употреблению ретро-игры, такие как *Doom* и *Blobwars*. **LXF**



➤ Если вы считали, что Linux — единственное удобное убежище, подумайте еще раз.

Booting.sh

По ряду технических причин нам не удалось сделать так, чтобы не-Linux операционные системы загружались прямо с DVD этого месяца. Но не переживайте, вы все равно можете с ними познакомиться, не устанавливая их. Мы включили скрипт **Booting.sh**, который запускает их в виртуальной машине Qemu. Чтобы он сработал, у вас должна быть установлена Qemu – вы найдете ее на диске или в репозиториях своего дистрибутива. В зависимости от вашего дистрибутива вы можете запустить его, или щелкнув по файлу и выбрав Run In A Terminal, или открыв терминал, перейдя на CD и введя

```
bash booting.sh
```

Перед вами появится список трех операционных систем, и вам просто надо будет выбрать ту, кото-

рую вы хотите загрузить. Все дополнительные параметры будут переданы в Qemu, поэтому

```
bash Booting.sh -full-screen
```

запустит выбранную операционную систему в полноэкранном режиме [Full Screen mode]. Если вы щелкните по окну Qemu, она захватит мышь. Чтобы освободить мышь, нажмите на Ctrl+Alt.

Если вы предпочитаете VirtualBox, а не Qemu, все ОС будут работать там с тем же успехом, но вам придется настроить виртуальные машины самостоятельно. В качестве альтернативы можете перенести образы диска на физический носитель и загрузиться с него. В Icaros имеется только образ DVD, так что вам придется записать его на диск. Haiku работает на образе Ahyboot (подобном Isohybrid в Linux),

поэтому его можно перенести на USB с помощью dd, или записать на USB. PC-BSD являет собой образ USB, поэтому ее можно только записать на брелок. Чтобы перенести образ Haiku или PC-BSD на устройство USB, используйте команду:

```
dd if=<path to image> of=/dev/sdX bs=4k
```

где <path to image> – размещение образа диска, а sdX – устройство USB. Если вы не испытываете особой уверенности, запустите **df -h**, чтобы вывести список дисков вашей системы. Все данные, уже имеющиеся на устройстве, будут стертты, поэтому вводите букву правильно. Введя не ту букву, можно лишиться всех данных с жесткого диска.

Пользователям Icaros придется записать образ на DVD (такая опция имеется и для Haiku).

Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам. Но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в «Линуксцентр» по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:

 <p>LXF155 Март 2012</p> <ul style="list-style-type: none">» Открытие Linux Как превратиться из салаги в гуру» Менеджерам на заметку Планировщики проектов» Dart Язык программирования от Google» Клевый музон Коллекция винтажных синтезаторов <p>LXFDVD: Ubuntu 11.10 LXF Remix и еще 6 загрузаемых дистрибутивов, CloneZilla, Condor Cloud, SELinux, игры и многое другое...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_155/ PDF-версия: shop.linuxformat.ru/elxf_155/</p>	 <p>LXF156 Апрель 2012</p> <ul style="list-style-type: none">» Raspberry Pi Компьютер-крошка делает революцию» Не стареют душой ветераны Продлеваем жизнь компьютерам» Dracut Инструмент ускорения загрузки, не зависящий от дистрибутива» Видеонаблюдение Сам себе Большой Брат <p>LXFDVD: Linux Gamers Live и еще 6 загрузаемых дистрибутивов, ZoneMinder, FontForge, FileMonitor и многое другое...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_156/ PDF-версия: shop.linuxformat.ru/elxf_156/</p>	 <p>LXF157 Май 2012</p> <ul style="list-style-type: none">» Linux — лидер Наша любимая ОС впереди планеты всей» Квартирка для проекта Какой хостинг-провайдер лучше» Node.js Альтернативная среда для JavaScript» Изучаем SQL База данных для телесериалов <p>LXFDVD: Sabayon 8, openSUSE 12.1 Cinnamon Remix и еще 2 загрузаемых дистрибутива, remastersys 3.0.0, DiffImage, IP-Link, PedSim и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_157/ PDF-версия: shop.linuxformat.ru/elxf_157/</p>
--	---	--

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала — оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru или shop.linuxformat.ru получают электронную версию в подарок!

Подписывайтесь на журнал на www.linuxformat.ru/subscribe/

Телефоны отдела подписки: Санкт-Петербург (812) 309-06-86, Москва (499) 271-49-54

Специальное предложение

Купите подборку журнала!

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски LXF. Если вы тоже этим озадачены, то в интернет-магазине «ГНУ/Линуксцентра» продолжается продажа журналов за 2010 год. Вы можете приобрести как отдельные номера изданий, так и подписки на 6 или 12 месяцев. Спешите — журналов осталось не так уж много!

shop.linuxformat.ru



Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
- » **имя_программы-1.0.1.tgz** – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

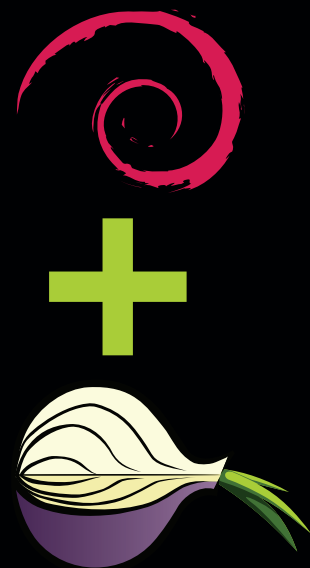
Внимательно прочтите это перед тем, как использовать LXF DVD!



CentOS 6.2

» Ядро 2.6.32 » Gnome 2.28 и KDE 4.3.4 » OpenJDK/КВМ » SELinux
32-разрядная сборка

А ТАКЖЕ: OracleJDK, средства обеспечения безопасности, полная совместимость с RHEL



Tails Linux

И три альтернативы Linux: » PC-BSD 9.0 » Напки Ft/Delphi 3 » Icaros Desktop 1.3

PC-BSD



А ТАКЖЕ: Audacity, FontForge, KTechLab и многое другое...

Июнь 2012
LXF DVD 158

LINUX
FORMAT

Июнь 2012
LXF DVD 158

LINUX
FORMAT

Содержание

LINUX FORMAT

ДИСТРИБУТИВЫ

CentOS 6.2 Установочный DVD, 32-разрядная сборка (вторая сторона **LXF DVD**)
Haiku R1/Alpha 3 Свободная ОС, созданная на базе BeOS (ISO-образ)
Isaros Desktop 1.3 Свободная ОС, совместимая с AmigaOS 3.1 на уровне API (ISO-образ)
PC-BSD 9.0 Вариант BSD UNIX для рабочих станций (ISO-образ)
Tails 0.1 (The Amnesic Incognito Live System) Live CD на базе Debian для сохранения приватности при работе в Интернет (загрузка с **LXF DVD**)

ПРОГРАММЫ НОМЕРА

Audacity 2.0.0 Один из наиболее популярных редакторов музыки
FontForge 2011.02.22 Программа для создания и редактирования шрифтов различных типов
KTechLab 0.3.7 Программа для моделирования работы электронных схем
Open 1.01 Система виртуализации, позволяющая эмулировать различные аппаратные архитектуры — ARM, MIPS, PowerPC, s390, SPARC и x86

НОТРИЦКС

Avision 0.7 Простая игра в стиле регро
Bluefish 2.2.1 Редактор текста, ориентированный на программистов
Din 2.0 Программа для генерации звуковых волн произвольной формы
Jskat 0.10.1 Популярная в Германии карточная игра Skat
Kenozoid 0.9.0 Программа планирования погружений для дайверов
MP3 Diag3 1.0.10 Средство проверки и восстановления поврежденных файлов MP3

Мир Linux

Mundus 11.3 Программа для удаления ненужных файлов
Muttrra 0.9.2 Генератор уворов и редактор шаблонов для их создания
Mylight 0.7.4.3 Графическое средство для работы с таблицами реляционных СУБД
SysUsage 5.2 Средство мониторинга системы с веб-интерфейсом
SubanLightScape 10.4 Редактор фотографий, позволяющий исправлять ошибки экспозиции при съемке в сложных условиях освещения

**Пожалуйста, перестаньте использовать
наименование Диска озвученное в журнале
с опубликованной в журнале
инструкцией!**

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru
ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех старых процессорах, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных в нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с нашего сайта: www.linuxformat.ru
 Издание: июль 2012 г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

Создание установочных дисков при помощи cdcrcord

Самый быстрый способ записать ISO-образ на чистую матрицу — это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке — метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа — это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?


Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.

**ФОРУМ
№1**
ДЛЯ ВСЕХ
ПОЛЬЗОВАТЕЛЕЙ
LINUX



LINUXFORUM.RU

alubuntu.ru



сообщество
пользователей
UBUNTU

iru
www.iru.ru

Возможности в надежности
Ноутбук iRU Patriot 510 15,6"



**На базе процессора Intel® Core™ i3
второго поколения**



- Эcran 15,6"
- WiFi
- Full HD TV
- Камера
- Web-камера

- Стильный внешний вид
- Компактный размер
- Низкое энергопотребление
- Идеальный мультимедиа центр

Intel, логотип Intel, Intel Inside, Intel Core и Core Inside являются товарными знаками корпорации Intel на территории США и других стран.

**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

**RED HAT ENTERPRISE
VIRTUALIZATION 3.0**

Red Hat и компания GNU/Линуксцентр приглашают принять участие в семинаре Red Hat Enterprise Virtualization 3.0



К регистрации приглашается широкий круг технических специалистов, перед которыми стоит задача создания собственных облачных структур и сервисов.

Red Hat Enterprise Virtualization для серверов представляет собой комплексное решение виртуализации, спроектированное для центров обработки данных и значительно увеличивающее эффективность капиталовложений и эксплуатации.

При использовании Red Hat Enterprise Virtualization для десктопов, пользовательские окружения находятся в виде виртуальных десктопов на серверах, расположенных в центре обработки данных. Пользователи подключаются к этим виртуальным десктопам либо с недорогих тонких клиентов, либо с устаревших ПК.

redhat
PREMIER BUSINESS PARTNER

Linux center
www.linuxcenter.ru

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ И ОБЯЗАТЕЛЬНАЯ РЕГИСТРАЦИЯ:
WWW.LINUXCENTER.RU/RHEV/

Unix Education Center
Россия, 190000, Санкт-Петербург
Черноморский переулок, дом 4
Тел.: +7 (812) 611-1575







**RHCSA/RHCE
Certification**

GNU/Линуксцентр приглашает на работу!



ВАКАНСИЯ: Drupal-разработчик

ОБЯЗАННОСТИ:

- » Создание сайтов с нуля на базе CMS Drupal: верстка, программирование, разработка баз данных, конвертация данных.
- » Разработка дополнительных модулей для CMS Drupal.
- » Внесение изменений в готовые скрипты.
- » Поддержка сайтов компании.

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

Школа LXF



Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Edumandriva: Свободное образование, и не только

Александр Казанцев оценивает путь, пройденный проектом со времени последнего обзора.



Наш эксперт

Александр Казанцев

Старший научный сотрудник ЗАО «Тырнет» и автор образовательного проекта Edumandriva в настоящий момент посвящает все свое время скрещиванию железа и софта под свободными лицензиями.

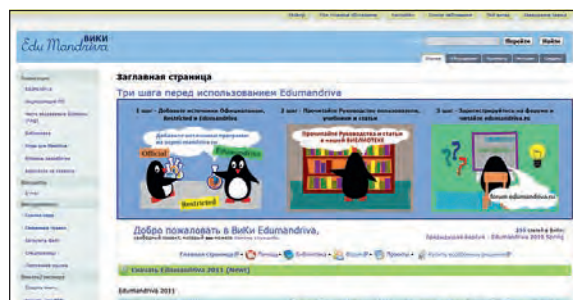
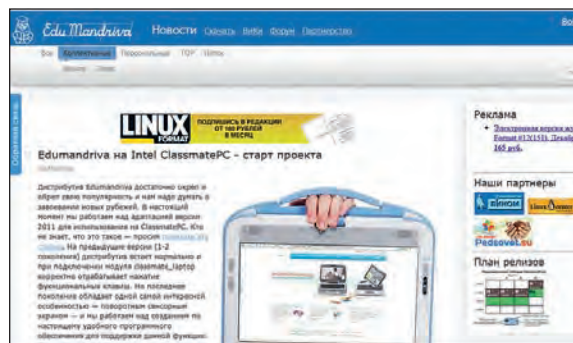
Проjekt Edumandriva уже не раз упоминался на страницах рубрики Школа LXF, и это не случайно. Сам проект родился примерно в то же время, что и сама рубрика, пережил ее закрытие и возрождение и является одним из долгожителей, которые уделяют внимание свободному и открытому ПО в России.

Знакомым с проектом будет интересно узнать, чем он живет сегодня, какие у него планы и что он может предложить для образования в настоящий момент. А тем, кто впервые услышал о проекте Edumandriva, будет интересно с ним познакомиться.

История проекта

Если прочитать любой из официальных пресс-релизов Edumandriva, то можно узнать, что EduMandriva – это образовательный проект компании Mandriva.Ru.

Цель проекта EduMandriva – создание образовательного пространства с использованием свободного и открытого программного обеспечения, нацеленного на консолидацию ресурсов образовательных учреждений (Центров компетенции Mandriva), социализацию пользователей и организацию исследований, разработки, внедрения и сопровождения методических, информационных, научных и программных инноваций – программного обеспечения на базе свободного и открытого ПО (на основе дистрибутива Mandriva GNU/Linux), обучающих курсов, методик, учебников, научных исследований.



► Сайт и Вики проекта EduMandriva.

Что такое Edumandriva

- » Одноименный репозиторий и сборник образовательного ПО для Mandriva Linux. В репозитории представлены программные продукты, адаптированные для школ России. Подключить репозитории можно на сайте <http://urpmi.mandriva.ru> или воспользовавшись инструкциями с wiki.edumandriva.ru.
- » Образовательный портал EduMandriva (<http://edumandriva.ru>). Здесь вы можете прочитать последние новости проекта, завести свой личный блог и обсуждать с коллегами насущные проблемы.
- » Образовательная инициатива Mandriva Russia – дистанционное и сертифицированное обучение

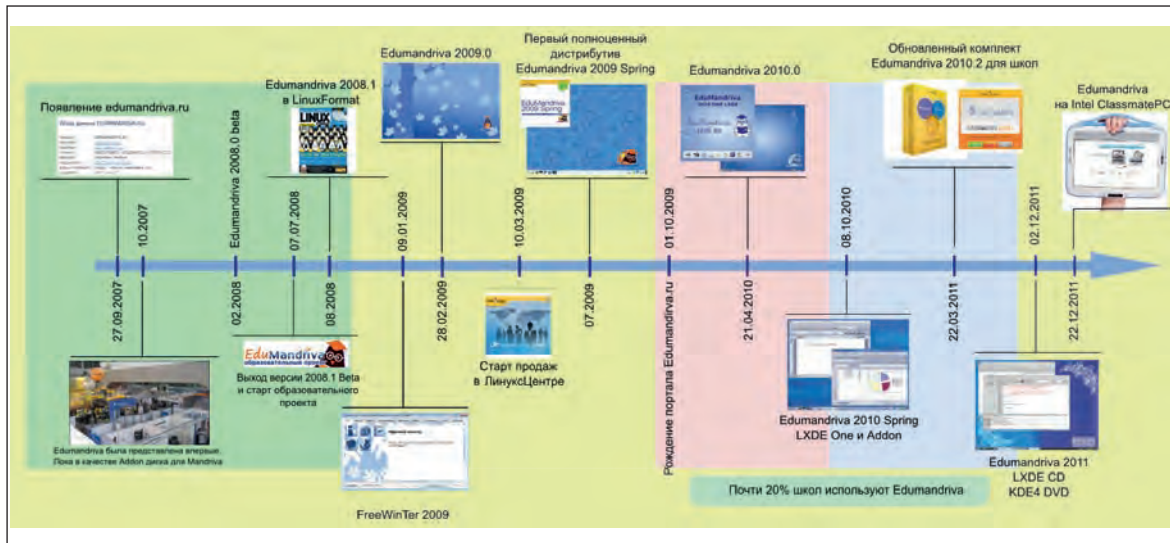
Mandriva (etraining.mandriva.ru). Курсы для пользователей и преподавателей.

- » Объединение исследовательских центров и Центров компетенции по свободному ПО.
- » Образовательные проекты с разработчиками и издателями учебников, методических пособий и учебных программ.
- » Форум и социальная сеть для общения пользователей – forum.edumandriva.ru. Задавайте вопросы и получайте ответы.
- » Энциклопедия свободного ПО и Библиотека. Справочник по свободным и открытым продуктам

(http://edumandriva.ru/wiki/index.php/Энциклопедия_ПО), а также методическая копилка статей, учебных пособий и учебников (<http://edumandriva.ru/wiki/index.php/Библиотека>).

О последних – поподробнее. Отсюда можно скачать релизы дистрибутива, посмотрев их состав, и получить агрегированный доступ к описаниям входящих программ (спасибо Википедии), а также пополняемой базе методических материалов по применению свободного ПО в образовании. В Библиотеке копятся ссылки на нашу рубрику «Школа LXF» – их общий объем уже близок к полноценной книге.

» На этой шкале времени представлены основные вехи развития проекта. Но интересное еще впереди.



На образовательном портале EduMandriva (<http://edumandriva.ru>) вы можете прочитать последние новости проекта, завести свой личный блог и обсуждать с коллегами насущные проблемы на форуме для общения пользователей, читать методические материалы и принимать участие в их разработке, загружать образовательное ПО, участвовать в его тестировании, локализации и разработке.

Проект начинался как простой репозиторий учебного ПО, но быстро развился в самостоятельное ответвление. И на протяжении последних лет он не только продолжал существовать, но и обзаводился все новыми и новыми возможностями и направлениями. На шкале времени вы можете увидеть основные вехи истории проекта.

И помните, что девиз проекта – «Сделано преподавателями для преподавателей».

Последние релизы школьного дистрибутива

Разработка школьного дистрибутива, как уже было сказано, является одним из приоритетных направлений проекта, и последние релизы, которые были выпущены, следующие (см. <http://edumandriva.ru/edumandriva/2012/01/11/predvaritelnye-obrazy-edumandriva-2011-h2-deyteriy.html>):

» **EduMandriva 2011 H2 LXDE CD** — легкая версия дистрибутива. Предназначена для относительно (от 256 Мб ОЗУ) машин и содержит в себе необходимый минимум программ:

Файловые менеджеры

- » **Midnight-Commander (mc)** Аналог FAR для консоли.

Архиваторы

- » Поддержка формата RAR.
- » Поддержка формата 7ZIP и русской кодировки в ZIP-архивах.

Офисные приложения

- » **LibreOffice Writer** Аналог MS Word, совместимый с ним по форматам файлов.
- » **LibreOffice Calc** Нетребовательные к ресурсам электронные таблицы – аналог Excel 97.
- » **ePDFviewer** Легкий просмотрщик PDF.

Растровые и векторные редакторы

- » **Gnome Paint** Легкий аналог MS Paint.
- » **Tuxpaint** Простой графический редактор для детей и взрослых.

Работа со звуком, музыкой и видео

- » **VLC** Популярный многофункциональный плеер.

Web/Интернет

- » **Mozilla Firefox** Один из ведущих web-браузеров.
- » **KompoZer** Легкий в освоении визуальный редактор.
- » **VpnPPTP** ПО для работы с MS VPN.

Программирование

- » **Basic256** Реализация BASIC для обучения программированию, совместимая с MS QBasic.
- » **Free Pascal** Компилятор Pascal и Object Pascal – в качестве IDE рекомендуется Lazarus или Geany.
- » **Geany** Простая IDE для FreePascal, C, C++ и т. п. + плагины.
- » **Scratch** Программирование для детей на основе Logo.
- » **Kumir** Алгоритмический язык программирования.

Клавиатурные тренажеры

- » **Klavaro** Клавиатурный тренажер по классической методике. Системы тестирования и дистанционного обучения
- » **iTest** Мощная клиент-серверная система составления вопросов и тестирования с учетом результатов.

Коннекторы

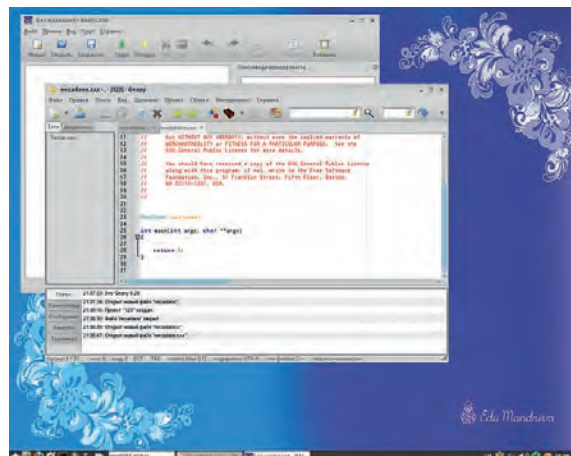
- » **Remmina** RDP/VNC/NX-клиент.

Системное ПО

- » **gparted** ПО для разбиения и управления разделами.
- » **gcalctool** Калькулятор.

Про большую часть программ можно найти информацию в Энциклопедии и прочитать в статьях Библиотеки (ну и, понятно, в предыдущих выпусках LXF).

» **EduMandriva 2011 H2 KDE4 DVD** – полноценный DVD с множеством прикладного ПО. К примеру, по сравнению с LXDE



» Внешний вид EduMandriva LXDE.

ClassmatePC & Edumandriva

Edumandriva на ClassmatePC является одним из немногих дистрибутивов, которые поддерживают данный класс устройств, и, наверное, единственным, который имеет почти 100% совместимость с линейкой (кроме самых последних версий). Причем поддерживаются не только

базовые функции, но и сенсорный экран с акселерометром, меняющий внешний вид рабочего стола в зависимости от применяемого режима. Подробнее об этом можно прочитать в заметке <http://edumandriva.ru/edumandriva/2011/12/12/edumandriva-na-intel-classmatepc---start-proekta.html>.

CD-версией стоит отметить добавление в образ пакетов Arduino, Blender, Scribus и других программ.

» **Edumandriva 2011 H2 CMPC DVD** – версия, созданная специально для компьютеров ClassmatePC (см. врезку вверху). В ней стоит отметить следующие моменты:

- » Поддержка оборудования ClassmatePC на Atom N-серии (3-е поколение).
- » Поддержка сенсорного экрана простым обновлением системы!
- » Рабочий стол LXDE.
- » Состав ПО аналогичен версии KDE4 DVD.

Свободная микроэлектроника для образования

Программирование – интересная вещь, но результат своего труда можно увидеть только на экране. С другой стороны, вещь, которую можно физически потрогать, вызывает больший интерес, чем виртуальная копия. Изучение алгоритмов дает понятие «исполнителя», что в большинстве детских умов ассоциируется с роботами или хотя бы электронными устройствами. Программируя реальные аппаратные устройства и взаимодействуя с ними, учащийся лучше понимает, для чего в принципе нужна программа и как работают вычислительные системы в целом. Также это просто интересно и познавательно, когда какая-то железка вдруг выполняет твои команды или персонаж на экране начинает слушаться только что собранной электронной схемы.

И здесь нам могут помочь свободное ПО и свободная электроника. Проект Arduino – как раз то, что позволит нам погрузиться в мир свободной электроники с минимальными затратами. Единственное «но» – программы для микропроцессоров (ATMega) используемых в Arduino (их еще называют скетчами) пишутся на ассемблере или используют специальные трансляторы с других языков. Но это уровень студентов вуза, и то продвинутого. Сама IDE Arduino представляет собой верх минимализма и аске-

тизма и требует знания языков уровня C или Java. Как это использовать для школьников?

Нам на помощь в этом случае приходит среда программирования «из кирпичиков» Scratch на основе языка Logo, с которой может справиться ребенок, начиная с 10–12 лет.

Объединив Scratch и Arduino, мы получим Scratchduino – программно-аппаратное решение, обладающее следующими положительными характеристиками:

- » легкая адаптация продукта для использования в учебных заведениях;
- » низкая стоимость и свободная лицензия;
- » использование как основы платформы свободной электроники Arduino, позволяющего расширять потенциал продукта;
- » наличие наработок по данной тематике в свободном доступе;
- » легкая расширяемость.

Казалось бы, при чем тут проект Edumandriva? На самом деле работа над данным проектом совместно с ЗАО «Тырнет» ведется опять же в рамках проекта Edumandriva. И основная информация, поддержка размещена на вики Edumandriva. Более того, дистрибутив Edumandriva является базовым для обкатки этого решения.

Так что же оно собой представляет? Если заглянуть на сайт (<http://edumandriva.ru/wiki/index.php/ScratchDuino>), то можно увидеть два основных направления: Scratchduino.Лаборатория и Scratchduino.Робоплатформа.

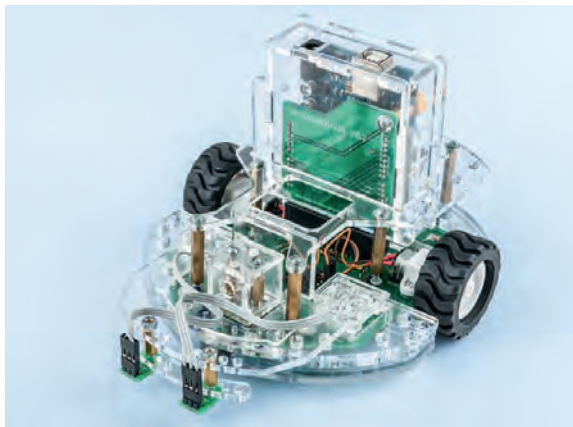
Первый представляет собой набор внешних датчиков ввода/вывода, таких как датчик освещения, звука, ползунок, кнопка, светодиоды и других, которые позволяют управлять и быть управляемыми с компьютера с помощью среды Scratch.

А с помощью подключаемых модулей расширения можно изучать живую природу, измерять температуру и даже построить простейший осциллограф.

Робоплатформа представляет собой колесную платформу, на которую с помощью магнитных креплений (без пайки и крепежных элементов) можно установить различные датчики и запрограммировать их работу, управляя получившимся роботом как по проводу, так и через Bluetooth-соединение.

Самое главное – это открытость проекта. Все схемы, разводка, схемы лазерной резки доступны и свободны для скачивания, поэтому никто не запрещает вам собрать для себя такие устройства. И вы можете быть уверены, что в рамках проекта Edumandriva мы будем рады, если вы предложите своего робота или какое-нибудь применение или улучшение для платы Лаборатории.

И опять же в Библиотеке Edumandriva вы можете скачать постоянно дополняемую и очень интересную книгу по робототехнике – http://edumandriva.ru/wiki/index.php/Библиотека_Edumandriva/Робототехника_с_пеленок. **LXF**



» Scratchduino.Лаборатория — наконец-то можно поуправлять компьютером в 4-м классе.



» Робот хоть и выглядит как окружность на колесиках, но обладает большими «взрослыми» возможностями.

Т е х н о л о г и я с ч а с т ь я



SUNRADIO.RU

сетевое радио под ключ на базе Linux
новое будущее вашей компании

pr@sunradio.ru | www.sunradio.ru

Ответ в конце задачника

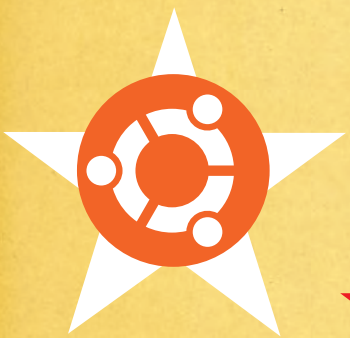
Для тех, кто хочет проверить себя, приводим итоговый скрипт из учебника «**LaTeX: Массивы макрокоманд**» на с. 88.

<code>\input{testsmakr}%Определения документа</code>	<code>\Qvery 9-й вопрос\endQvery</code>
<code>\kafedra{название}</code>	<code>\Qvery 10-й вопрос\endQvery</code>
<code>\zavkaf{Ф.И.О.}</code>	<code>\printQ{6} %Вывод на печать 6-го вопроса</code>
<code>\datautv{01.01.2012}</code>	<code>\makelistqvery{} %Создать список вопросов</code>
<code>\vuz{название}</code>	<code>\newpage %Билет с помощью окружения EkzBilet</code>
<code>\bilettitle{Билет}</code>	<code>\begin{EkzBilet}</code>
<code>\biletauthor{Ф.И.О.}</code>	<code>\printQ{4}</code>
<code>\predmet{предмет} \spez{00.00.00}</code>	<code>\printQ{7}</code>
<code>\begin{document}</code>	<code>\printQ{1}</code>
<code>%Определяем сами вопросы, которые при этом на печать не выводятся</code>	<code>\printQ{4}</code>
<code>\Qvery 1-й вопрос\endQvery</code>	<code>\end{EkzBilet}</code>
<code>\Qvery 2-й вопрос\endQvery</code>	<code>%Билет с помощью команды \printbilet с теми же вопросами</code>
<code>\Qvery 3-й вопрос\endQvery</code>	<code>\printbilet{1}\and{7}\and{1}\and{4}}</code>
<code>\Qvery 4-й вопрос\endQvery</code>	<code>\newpage %блок из двух билетов</code>
<code>\Qvery 5-й вопрос\endQvery</code>	<code>\twotestblok{4}\and{1}}{5}\and{4}}</code>
<code>\Qvery 6-й вопрос\endQvery</code>	<code>\newpage %блок из четырех билетов</code>
<code>\Qvery 7-й вопрос\endQvery</code>	<code>\fiertestblok{8}\and{3}}{9}\and{2}}{2}\and{3}}{3}\and{2}}</code>
	<code>\end{document}</code>

**Пропустили номер? Узнайте на с. 104,
как получить его прямо сейчас!**



LXF ИМЕЕТ ЧЕСТЬ СООБЩИТЬ



UBUNTU

★★ ПРОТИВ ★★

WINDOWS



В МАТЧЕ НА НАСТОЛЬНОМ ПК

В июльском номере

Дра-а-а-чка!

Две зрелых настольных операционных системы – и обе пытаются капитально реформироваться и апеллировать к новичкам. Запахло разборками...

Ведите себя хорошо

Пишите сообщения об ошибках так, чтобы разработчикам не было мучительно больно за бесцельно подавленный порыв плюнуть вам в личико.

...Raspberry Pi

Если мы дорвемся хоть до одного экземпляра, то уж власть исследуем его возможности. Arch Linux, дело за тобой!

Содержание последующих выпусков может меняться без уведомления – чего доброго, нас повяжут за использование Тор.

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Илья Аввакумов, Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова, Татьяна Цыганова

Редактор диска

Кирилл Степанов

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатели

Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Ланинь»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 4822

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон (Graham Morrison) graham_morrison@futurenet.com

Заместитель редактора Эндрю Грегори (Andrew Gregory)

agregory@futurenet.com

Штатный автор Джонатан Робертс (Jonathan Roberts)

jonathan.roberts@futurenet.com

Редактор диска Бен Эверард (Ben Everard) ben_everard@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

(Efraim Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.com

Выпускающий редактор Гэри Уокер (Gary Walker) gary.walker@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Иэн Бартон (Ian Barton), Нейл Ботвик (Neil Bothwick), Крис Браун (Chris Brown), Алекс Кокс (Alex Cox), Бен Эверард (Ben Everard), Грэм Моррисон (Graham Morrison), Джонатан Робертс (Jonathan Roberts), Маянк Шарма (Mayank Sharma), Шашанк Шарма (Shahank Sharma), Майк Сондерс (Mike Saunders), Ник Вейч (Nick Veitch), Евгений Балдин, Андрей Боровский, Артем Зорин, Александр Казанцев, Андрей Ушаков, Алексей Федорчук

Художественные ассистенты Стэйси Блэк (Stacey Black),

Кейт Мак-Доннелл (Kate McDonnell)

Иллюстрации Шейн Коллиндж (Shane Collinge), Крис Хедли (Cris Hedley),

Ely Walton Illustrations, iStockPhoto, Саймон Мидлвик (Simon Middleweek)

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Tel. +44 01225 442244 Email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):

Лиговский пр., 50, корп. 15

Тел. +7 (812) 309-06-86

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЗИ)

Тел./факс +7 (499) 271-49-54

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензируются Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

LINUX – зарегистрированный товарный знак Линуса Торвальдса (Linus Torvalds). «GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журнале, издаваемом Future plc group company, обращайтесь на сайт <http://www.futureplc.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



Mandriva Linux — один из самых популярных дистрибутивов GNU/Linux в мире. Главные преимущества Mandriva — дружелюбный интерфейс, простота настройки, возможность быстрой адаптации пользователей, ранее не знакомых с этой ОС, совместимость с широким спектром программного и аппаратного обеспечения.

Корпоративные продукты Mandriva Linux

Mandriva 2010.2 Powerpack

Дистрибутив Mandriva 2010.2 Powerpack включает набор офисных и серверных приложений, и подходит для установки на офисной или домашней рабочей станции и на сервере. Дружелюбный интерфейс, простота настройки Mandriva Powerpack, совместимость с широким спектром аппаратного обеспечения и совместимость с «1С:Предприятие» обеспечивают корпоративным пользователям возможность легкого перехода с Windows на GNU/Linux.

Mandriva Enterprise Server 5

Mandriva Enterprise Server 5 (MES 5) — это надежный и производительный дистрибутив GNU/Linux для корпоративного сервера. MES 5 поможет вам снизить текущие расходы и упростить инфраструктуру. В MES 5 интегрированы серверные разработки программистов Mandriva, в том числе сервер каталогов пользователей Mandriva Directory Server, а также ведущие свободные серверные приложения, которые помогут вам с минимумом затрат времени и энергии настроить и поддерживать необходимые вам серверы. Срок поддержки дистрибутива — 5 лет.

Сертифицировано ФСТЭК

Дистрибутивы Mandriva Linux сертифицированы по требованиям ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ, что дает возможность использовать их для обработки конфиденциальной информации в автоматизированных системах класса до 1Г включительно и обработки персональных данных в информационных системах класса до К2 включительно.

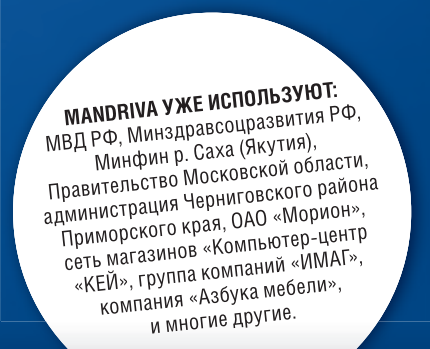
- **Mandriva 2008 Spring Powerpack** — дистрибутив для рабочей станции или небольшого сервера.
- **Mandriva Corporate Server 4 Update 3** — дистрибутив для создания корпоративного сервера.
- **Mandriva Flash** — дистрибутив GNU/Linux, загружающийся и работающий прямо с USB-носителя.

EduMandriva — свободное ПО для образования

- Создано с участием российских преподавателей и методистов.
- Все ПО, необходимое для преподавания информатики.
- Методические материалы.

Наименование	Стоимость, руб.
Корпоративные продукты Mandriva	
Mandriva Linux 2010.2 Powerpack (DVD-Box)	1 300
Услуга подписки на Mandriva Enterprise Server 5 на 1 год, базовый уровень (с физическим носителем)	13 300
Услуга подписки на Mandriva Enterprise Server 5 на 3 года, базовый уровень (с физическим носителем)	34 800
Продукты Mandriva для образования	
Комплект программного обеспечения Mandriva Linux и EduMandriva для школ	3 500
Сертифицированные ФСТЭК продукты Mandriva	
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 10 рабочих мест	28 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 5 рабочих мест	14 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 1 рабочее место	4 990
Сертифицированный ФСТЭК Mandriva Corporate Server 4.0 Update 3	10 050

С вопросами по продуктам Mandriva обращайтесь в «ГНУ/Линуксцентр»!



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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





ВАМ ПОНРАВИТСЯ ЭТИ ПРЕДЛОЖЕНИЯ

HETZNER DEDICATED ROOT SERVER EX 4S

- Intel®Core™ i7-2600 Quadcore с поддержкой технологии Hyper-Threading
- 32 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 5500 рублей

2200
рублей в месяц

HETZNER DEDICATED ROOT SERVER EX 8

- Intel®Xeon® E3-1275 Quadcore с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM ECC
- До четырёх жёстких дисков на выбор
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- KVM-over-IP для постоянного доступа
- Установка из образов
- Без минимального контракта
- Установка 5500 рублей

от **3500**
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